

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture

Présentée et soutenue par :

M. MOHAMED MAACHAOUI

le vendredi 12 juin 2015

Titre :

SECURITE ET PERFORMANCES POUR LES RESEAUX DE NOUVELLE GENERATION (NGN)

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

M. CHRISTIAN FRABOUL

M. ANAS ABOU EL KALAM

Rapporteurs :

Mme MARYLINE LAURENT, TELECOM SUD PARIS

M. MOHAMED MOSBAH, UNIVERSITE BORDEAUX 1

Membre(s) du jury :

M. VINCENT NICOMETTE, INSA TOULOUSE, Président

M. ABDELLAH AIT OUAHMAN, UNIV. DE CADI AYYAD MARRAKECH MAROC, Membre

M. ANAS ABOU EL KALAM, INP TOULOUSE, Membre

M. CHRISTIAN FRABOUL, INP TOULOUSE, Membre

M. JUAN LOPEZ, AIRBUS FRANCE, Membre

M. SALAH EDDINE LABHALLA, UNIV. DE CADI AYYAD MARRAKECH MAROC, Membre

Avant-Propos

Les travaux présentés dans ce mémoire ont été effectués au sein du Laboratoire IRIT (Institut de Recherche en Informatique de Toulouse) et le laboratoire OSCARS (Laboratoire d'optimisation des systèmes de communications avancés, Réseaux et Sécurité).

Je tiens à remercier tout d'abord Michel Daydé directeur de l'IRIT pendant mon séjour, de m'avoir permis de mener mes recherches dans ce laboratoire.

Ma reconnaissance se tourne particulièrement vers André-Luc Beylot, ancien responsable de l'équipe Ingénierie Réseaux et Télécommunications (IRT) et son successeur Jean-Luc Scharbarg, pour m'avoir accueilli dans cette équipe de recherche.

Mes plus grands remerciements sont naturellement pour Christian Fraboul, Anas Abou El Kalam et Abdellah Ait Ouahman, qui m'ont encadré tout au long de ma thèse. Ma considération est inestimable. Leurs remarques et critiques pertinentes m'ont conduit vers la bonne voie. Leur soutien m'a permis de ne jamais faiblir et de poursuivre toujours plus loin mes travaux. Je tiens également à souligner que la confiance qu'ils ont mise en moi a été un moteur à ma réussite.

J'exprime ma gratitude à Vincent Nicomette, Professeur au LAAS, pour l'honneur qu'il me fait en présidant mon Jury de Thèse, ainsi qu'à :

- Mme. Maryline Laurent, Professeur Telecom Sud Paris
- M. Mohamed Mosbah, Professeur LaBRI/INP Bordeaux
- M. Salah Eddine Labhalla, Professeur Université Cadi Ayyad
- M. Juan Lopez, Network Expert Airbus Aircraft

pour l'honneur qu'ils me font en participant à mon jury. Je remercie particulièrement Maryline Laurent et Mohamed Mosbah qui ont accepté la charge d'être rapporteur.

Je remercie tout l'équipe IRT, les permanents, les doctorants et les stagiaires.

Mes remerciements s'adressent également à l'ensemble des services de l'IRIT, qui permettent à chacun de travailler dans les meilleures conditions.

Il m'est agréable de remercier chaleureusement tous ceux qui, en dehors du laboratoire, m'ont accompagné et soutenu. Je pense particulièrement à mon épouse Sana qui m'a beaucoup aidé et qui a partagé avec moi des moments faciles et difficiles durant ces années de thèse.

Ces avant-propos seraient incomplets sans un remerciement adressé aux membres de ma famille, en particulier mes parents. Ce travail leur appartient à tous.

Je pense également à mes amis Hassan, Abdeljebar, Khalid, Mahmoud, Hicham, Ahmed, Farouk, ma tante Somia et mon cher professeur Noureddine ainsi que tous les autres gens aimables et serviables qui m'ont soutenu et qui ont contribué à mon enrichissement personnel.

À tous ces gens-là, je serais éternellement reconnaissant. Merci.

Résumé

L'IMS (IP Multimedia Subsystem) constitue l'architecture clé de contrôle pour les réseaux de nouvelle génération (NGN : Next Generation Network). IMS offre aux opérateurs réseaux la possibilité d'étendre leurs services, en intégrant la voix et des communications multimédia et de les livrer dans de nouveaux environnements avec de nouveaux objectifs. Sa sécurité totale mais à moindre coût est donc primordiale, principalement l'authentification. En IMS l'authentification est divisée en deux phases, une au niveau du domaine PS (Packet-Switch) avec le protocole 3GPP-AKA, et l'autre au niveau IMS en utilisant le protocole IMS-AKA.

Dans notre première contribution, nous proposons un nouveau protocole d'authentification plus sécurisé que celui utilisé en IMS (IMS-AKA) et plus performant en termes d'utilisation de la bande passante et de temps de traitement. Notre méthode d'analyse repose sur la quantification de la signalisation induite par l'authentification IMS. La quantification est effectuée à l'aide d'expérimentations réelles. Sur la base des résultats obtenues, nous pouvons confirmer que notre protocole (1) peut économiser au moins 21,5% du trafic SIP/Cx par rapport à l'IMS-AKA, (2) permet de réduire la consommation de la bande passante de 27% par rapport à l'IMS-AKA, (3) résiste aux attaques atteignant la confidentialité et l'intégrité des données lors d'un enregistrement IMS (validé par AVISPA).

Dans notre seconde contribution, nous avons présenté un nouveau modèle, nommé virtual walled-garden, de fourniture de services centré sur l'utilisateur en IMS. Ce modèle de fourniture de service permet d'offrir plus de liberté d'utiliser les services de tout fournisseur de contenu en fonction des besoins et préférences des utilisateurs. De cette manière les trois parties (utilisateur, fournisseurs de services et opérateur IMS) sont satisfaites. Les utilisateurs auront accès à un plus large éventail de services soutenus par l'IMS, les fournisseurs de services peuvent mettre en œuvre un large éventail de services IMS/SIP sans aucun investissement sur la mise en œuvre d'un réseau de cœur IMS ou de sa maintenance. Quant aux opérateurs cette façon de faire constitue une nouvelle forme de partenariat d'affaires avec les fournisseurs de services. Le modèle virtual walled-garden se base sur une fédération d'identité multi niveaux pour prendre en considération plusieurs niveaux de sécurité selon la criticité des applications sollicitées.

Abstract

The IMS (IP Multimedia Subsystem) architecture is the key control for next generation networks (NGN). IMS gives network operators the opportunity to extend their services, including voice and multimedia communications and deliver them in new environments with new goals. Its security is paramount, especially authentication. In IMS, authentication is divided into two phases a PS (Packet-Switch) domain-level with the 3GPP-AKA protocol, and a second at IMS level using the IMS-AKA protocol.

In our first contribution, we propose a new IMS authentication mechanism that improves the IMS-AKA in terms of security and more efficient in the use of bandwidth and processing time. Based on the results obtained, we can confirm that our protocol can save at least 21.5% of SIP/Cx traffic compared to the IMS-AKA and resists to attack reaching the confidentiality and integrity of data in an IMS registration (validated by AVISPA).

In our second contribution, we propose a new Service provisioning model: Virtual Walled-Garden. This new model allows the user accessing all the applications, even the external ones transparently, simulating a walled-garden environment. This model will create a trust link between IMS domain and external services, and will reduce the burden of both end users and SPs through a Single Sign-On (SSO) feature, using identity federation. We also introduce the notion of security level to classify the SPs in a Multi-level model.

Table des matières

Résumé ... 1

Abstract ... 5

Table des matières ... 7

Liste des figures ... 13

Liste des tableaux ... 15

Introduction générale ... 17

Chapitre 1 : Architecture IMS ... 21

1.1. Motivation pour l'architecture IMS ... 21

1.2. Relations entre NGN et IMS ... 22

1.2.1. Définition des NGN ... 23

1.2.2. Architecture des réseaux NGN ... 23

1.2.3. Types de NGN ... 25

1.3. IMS : IP Multimedia Subsystem ... 26

1.3.1. Définition ... 26

1.3.2. Bénéfices de l'architecture IMS ... 28

1.3.2.1. Une plateforme commune ... 29

1.3.2.2. La qualité de service ... 29

1.3.2.3. La facturation ... 29

1.3.2.4. Une architecture IP de bout en bout ... 30

1.4. Architecture IMS ... 30

1.4.1. Architecture fonctionnelle en couche d'IMS ... 30

1.4.1.1. La couche physique (accès) ... 31

1.4.1.2. La couche transport ... 32

1.4.1.3. La couche contrôle de session (contrôle) ... 32

1.4.1.4. La couche application ... 32

1.4.2. Principaux composants de l'architecture IMS ... 33

1.4.2.1. Le Home Subscriber Server (HSS) ... 33

1.4.2.2. CSCF: Call Session Control Function ... 33

1.4.2.2.1. Le Proxy CSCF ... 34

1.4.2.2.2. L'Interrogating CSCF ... 34

1.4.2.2.3. Le Serving CSCF ... 34

1.4.2.3. Serveurs d'applications ... 35

1.4.2.3.1. SIP AS ... 35

1.4.2.3.2. IM-SSF ... 36

1.4.2.3.3. OSA-SCS ... 36

1.4.3. Les interfaces ... 36

1.4.3.1. Point de référence Gm : ... 37

1.4.3.2. Point de référence Mw ... 37

1.4.3.3. Point de référence ISC: ... 37

1.4.3.4. Point de référence Cx : 37
1.4.3.5. Point de référence Dx 38
1.4.4. Gestion des identités 38
1.4.4.1. Public User Identity 38
1.4.4.2. Private User Identity 39
1.4.4.3. Relations entre Public et Private User Identity 39
1.4.5. Carte USIM et ISIM 40
1.4.5.1. USIM (Universal Subscriber Identity Module) 40
1.4.5.2. ISIM (IMS Subscriber Identity Module) 41
1.4.6. Signalisation en IMS 41
1.5. Principaux protocoles 42
1.5.1. Le protocole SIP 42
1.5.1.1. Responsabilités du SIP : 42
1.5.1.2. Type de signalisation SIP : 43
1.5.1.3. Format des messages SIP 43
1.5.2. Le protocole Diameter 45
1.5.2.1. Les responsabilités de Diameter 45
1.5.2.2. Sa structure de message 46
1.6. Conclusion 47
Chapitre 2 : Sécurité en IMS 49
2.1. Défis de sécurité en IMS et attaques potentielles 49
2.2. Mécanisme de sécurité en IMS 55
2.2.1. Sécurité des accès 56
2.2.1.1. Authentification 56
2.2.1.2. Tunnel IPSec entre terminal et P-CSCF 57
2.2.2. Sécurité dans le réseau de cœur : Architecture NDS 57
2.2.2.1. Passerelles de sécurité 57
2.2.2.2. Fonction I-CSCF (THIG) 59
2.2.3. Framework de sécurité proposé par le 3GPP2 59
2.2.3.1. Objectifs des solutions de sécurité IMS 59
2.2.3.2. Associations de sécurité 61
2.3. Conclusion 62
Chapitre 3 : Mécanismes d'authentification 65
3.1. Notion d'authentification 66
3.1.1. Définitions 66
3.1.2. Facteurs d'authentification 67
3.2. Principes cryptographiques 68
3.2.1. La cryptographie symétrique 68
3.2.2. La cryptographie asymétrique 69
3.2.3. Notion de certificat 70
3.2.4. Structure d'un certificat 70
3.2.5. Signatures de certificats 72
3.2.6. Exemple d'utilisation conjointe de mécanismes d'authentification : IPsec 72
3.3. Classification des méthodes d'authentification 73

3.3.1. Authentification par mots de passe statiques 73
3.3.2. Authentification par procédés biométriques 73
3.3.3. Authentification par mots de passe dynamiques 74
3.3.4. Protocole défi-réponse 74

3.4. Conclusion 76

Chapitre 4: The PKI-based One-way IMS-AKA 77

4.1. Procédure d'enregistrement dans IMS : Requête REGISTER 78
4.1.1. Prérequis pour obtenir le service IMS 78
4.1.2. Connectivité IP en utilisant GPRS 81
4.1.2.1. Déroulement global : 81
4.1.2.2. Attachement au réseau GPRS : 81
4.1.3. Découvert du P-CSCF 83
4.1.4. Enregistrement et authentification IMS 84
4.1.4.1. Enregistrement IMS avec une ISIM 84
4.1.4.2. Enregistrement IMS avec une USIM 89
4.1.4.3. GPP two-pass authentication 91

4.2. Travaux connexes 93
4.2.1. Approche 1 : One-Pass Authentication 93
4.2.2. Approche 2 : One-Pass AKA 94
4.2.3. Analyse de sécurité: 95

4.3. Protocole proposé: PKI-based One-Way IMS-AKA 96
4.3.1. Protocole proposé version I: Utilisation du Diffie-Hellman 96
4.3.2. La version améliorée du protocole proposé : PKI-based One-way IMS-AKA 97

4.4. Implémentation 101
4.4.1. Outils utilisés 101
4.4.1.1. Little IMS 101
4.4.1.2. Cipango 103
4.4.1.3. Maven 103
4.4.1.4. Outils cryptographiques 104
4.4.2. Détails de l'implémentation 105

4.5. Analyse de performances et de sécurité 107
4.5.1. Analyse de sécurité 108
4.5.2. Analyse de coût du nombre de messages échangés 111
4.5.3. Analyse de la consommation de la bande passante : 113
4.5.3.1. Analyse de la bande passante pour IMS-AKA 113
4.5.3.2. Analyse de la bande passante pour le protocole proposé 114
4.5.3.3. Comparaison 115

4.6. Tests d'évaluations des performances 115
4.6.1. Scenario 1: Temps de traitement dans le réseau de cœur IMS 117
4.6.2. Scenario 2: comparaison du délai entre deux réseaux d'accès : filaire et sans-fil 119

4.7. Mécanismes de vérification et de révocation des certificats 122
4.7.1. OCSP (*Online Certificate Status Protocol*) 122
4.7.2. 2-3 CRT (*Certificate Revocation Tree*) 122
4.7.3. Les CRLs 123
4.7.3.1. Delta-CRL : 123

4.7.3.2. CRL Distribution Points : 123
4.7.3.3. Over-issued CRL : 123
4.7.4. Comparaison des mécanismes de vérification de révocation de certificats 124
4.7.4.1. Estimation des coûts 125
4.8. Conclusion 127
Chapitre 5: Virtual Walled-Garden Model for IMS Services Provisioning 129
5.1. Architecture et principes pour la mise en œuvre des services 130
5.1.1. Déclenchement des services sur la base de filtres 132
5.1.2. Identité des services 132
5.1.3. Configuration des services 132
5.1.4. Exemple d'établissement de session 134
5.2. Défis de fourniture de services en IMS 136
5.3. Single Sign On : SSO 139
5.3.1. Qu'est-ce que le SSO ? 139
5.3.2. Objectifs d'un système de Single Sign-On 140
5.3.2.1. Apport ergonomique pour l'utilisateur 140
5.3.2.2. Amélioration de la sécurité 140
5.3.3. Les types d'authentification SSO 141
5.3.4. Les composantes SSO 142
5.3.5. Les différentes approches de SSO 143
5.3.5.1. L'approche centralisée 143
5.3.5.2. L'approche fédérative 144
5.3.5.3. L'approche coopérative 146
5.3.6. OpenID et OAuth 146
5.3.6.1. OpenID 146
5.3.6.2. OAuth 147
5.3.6.3. Utilisation de OpenID conjointement avec OAuth 148
5.3.6.4. Flux d'autorisation OpenID + OAuth 148
5.3.7. SAML 150
5.3.7.1. Profiles SAML 151
5.3.7.2. Sécurité en SAML 152
5.4. Virtual walled-garden 152
5.4.1. Approche SSO choisie 153
5.4.2. Intégration des mécanismes SSO en IMS 154
5.4.3. Etablissement de la fédération 157
5.4.3.1. Introduction à la confiance 157
5.4.3.2. La délégation des tâches administratives 157
5.4.3.3. L'ouverture des systèmes d'informations 157
5.4.3.4. Sécurisation des architectures orientées services 158
5.4.3.5. Identités locales d'utilisateurs 158
5.4.3.6. Procédure de construction de la fédération 158
5.4.3.7. Protection de la vie privée 160
5.4.4. SIP SAML : Profil et binding 161
5.4.4.1. SAML basé sur HTTP-URI pour un binding avec le protocole SIP 161
5.4.4.2. SAML basé sur SIP-URI en utilisant des messages SOAP sur SIP 163
5.4.5. SSO Multi-niveaux 164

5.4.5.1. Amélioration d'IdP 165
5.4.5.2. Les stratégies d'authentification LOA 165
5.4.5.3. Exemple d'implémentation d'un système avec deux niveaux LoA : 166
5.5. Conclusion 167
Conclusions et perspectives 169
Liste des publications 173
Références 175

Liste des figures

Figure 1.1. Offre une représentation communément admise de l'architecture des réseaux NGN *24*
Figure 1.2. Vue simplifiée d'IMS par TISPAN [Etsi11] *27*
Figure 1.3. Vue simplifiée d'IMS par CISCO *28*
Figure 1.4. Architecture de l'IMS *31*
Figure 1.5. Points de référence en infrastructure IMS *37*
Figure 1.6. Relation entre l'identité privée et publiques en IMS 3GPP R5. ..... *39*
Figure 1.7. Relation entre l'identité privée et publiques en IMS 3GPP R6. ..... *40*
Figure 1.8. Format d'un message SIP *44*
Figure 1.9. Exemple de requête SIP d'invitation *45*
Figure 1.10. Niveaux du protocole Diameter (base et application) *46*
Figure 1.11. Structure du message Diameter *47*
Figure 2.1. Les menaces potentielles en IMS *50*
Figure 2.2. L'attaquant envoie un message BYE pour mettre fin à la connexion *53*
Figure 2.3. Exemple d'un dépassement de tampon dans le cadre du protocole SIP *54*
Figure 2.4. Exemple d'une injection SQL dans le cadre du protocole SIP *54*
Figure 2.5. Exemple d'un scénario de vol de médias *55*
Figure 2.6. Architecture de passerelle de sécurité *58*
Figure 2.7. Sécurité dans le réseau de cœur IMS *58*
Figure 2.8. Vue globale de la sécurité en IMS *61*
Figure 2.9. Association de sécurité IMS *62*
Figure 3.1. Format d'un certificat. *71*
Figure 3.2. Vérification d'un certificat *72*
Figure 4.1. Prérequis pour obtenir le service IMS *79*
Figure 4.2. Prérequis pour obtenir le service IMS (cas du GPRS comme réseau d'accès) *80*
Figure 4.3. Obtention de la connectivité IP en GPRS *81*
Figure 4.4. Le flux de messages pour l'authentification 3GPP (niveau GPRS) *82*
Figure 4.5. Processus d'authentification/enregistrement en IMS avec le protocole IMS-AKA *88*
Figure 4.6. Exemple d'une requête SIP REGISTER (Message 11) [Gonz00] *89*
Figure 4.7. Structure du IMSI *90*
Figure 4.8. Enregistrement illégal au réseau IMS. *92*
Figure 4.9. Processus du protocole "One-Pass Authentication" *94*
Figure 4.10. Processus du protocole "The Proposed One-Pass AKA" *95*
Figure 4.11. La requête SIP REGISTER envoyé par le terminal IMS *99*
Figure 4.12. Protocole proposé : version basée sur les PKI *101*
Figure 4.13. Les composants du LittleIMS *102*
Figure 4.14. Le champ "Authorization" *105*
Figure 4.15.Wireshark Capture *106*
Figure 4.16. Le flux de messages SIP au niveau du I-CSCF visualisé sur l'interface web du I-CSCF *107*
Figure 4.17. Les identités enregistrées visualisées sur l'interface web du HSS *107*
Figure 4.18. Le modèle AVISPA utilisé pour vérifier le protocole PKI-based one-way IMS-AKA *109*
Figure 4.19. Amélioration du protocole proposé par rapport à IMS-AKA *112*
Figure 4.20. Architecture utilisée pour les tests *116*
Figure 4.21. Temps de traitement au niveau du réseau de cœur *118*

(a) échelle linéaire, (b) échelle logarithmique *118*
Figure 4.22. Délai d'authentification pour un réseau d'accès filaire (Ethernet) *120*
(a) échelle linéaire, (b) échelle logarithmique *120*
Figure 4.23. Délai d'authentification pour un réseau d'accès sans-fil *121*
(a) échelle linéaire, (b) échelle logarithmique *121*
Figure 5.1. Architecture de référence pour le support des services IMS *131*
Figure 5.2. Configuration des services IMS via l'interface « Ut » entre le terminal et le serveur d'application .. *133*
Figure 5.3. Mise en communication de deux terminaux *135*
Figure 5.4. Le modèle Virtual Walled-Garden *139*
Figure 5.5. Architecture Web SSO *141*
Figure 5.7. Etablissement d'une session pour une application utilisant SSO *143*
Figure 5.8. Approche SSO centralisée *144*
Figure 5.9. Approche SSO fédérative *145*
Figure 5.10. Authentification basée sur OpenID + OAuth *148*
Figure 5.11. Architecture de base SSO pour IMS *155*
Figure 5.12. L'architecture IMS avec l'IdP dans le réseau IMS *156*
Figure 5.13. Fédération d'identité avec Pseudonyme persistante *160*
Figure 5.14. Séquence SSO initiée par l'IdP pour des applications IMS (idP à l'intérieur du réseau IMS) *162*
Figure 5.15. Assertion SAML *163*
Figure 5.16. Exemple multi-niveau avec deux niveaux LoA *166*
Figure 5.17. Authentification avec Diffie-Hellman utilisant la signature numérique pour IMS *168*

Liste des tableaux

Tableau 1.1. Liste des différentes méthodes SIP *44*
Tableau 1.2. Liste des différents types de codes retours *45*
Tableau 2.1. Dimensions de sécurité et menaces correspondantes *51*
Tableau 4.1. Etapes identiques entre GPRS et IMS authentification *91*
Tableau 4.2. Interfaces Web de configuration *103*
Tableau 4.3. Caractéristiques matérielles et logicielles de la machine "cœur IMS" *116*
Tableau 4.4. Caractéristiques matérielles et logicielles de la machine "client IMS" *117*
Tableau 4.5. Temps moyen du traitement dans le réseau de cœur *119*
Tableau 4.6. Critères de comparaison *124*
Tableau 4.7. Définition des coûts *125*
Tableau 4.8. Coût de mise à jour *126*
Tableau 4.9. Coût de vérification *126*
Tableau 4.10. Comparaison entre le protocole proposé et d'autres approches discutées dans ce chapitre *128*
Tableau 5.1. Exemple d'un système d'authentification avec deux niveaux LoA *167*

Introduction générale

La libéralisation du secteur des télécommunications, les nouvelles demandes en services innovants pour répondre aux besoins des utilisateurs, et l'explosion du trafic numérique vu l'augmentation de l'utilisation d'Internet ont obligé les opérateurs Télécoms de chercher à optimiser leurs infrastructures. Cette optimisation passe à travers la réduction de leurs CAPEX (*CAPital EXpenditure* ou budget d'investissement) et OPEX (*OPerating EXpenditure* ou budget d'exploitation). Par voie de conséquence ils ont investi en de nouvelles technologies basées sur le protocole IP qui permettent la mutualisation des ressources réseaux et des services. Le concept des réseaux de nouvelles générations, baptisé NGN (*Next Generation Network*), est alors né. L'introduction des réseaux NGN comporte des aspects économiques et techniques. Sur le plan économique, elle permet d'augmenter la productivité en créant des nouveaux usages en fonction des préférences de l'utilisateur. Elle permet également la réduction des coûts par la création des infrastructures, avec un seul type de réseau de transport, au lieu de celles spécifiques à chaque réseau d'accès. Techniquement, l'architecture NGN rend le réseau souple dans la mesure où elle permet la définition et l'introduction facile de nouveaux services à forte valeur ajoutée.

L'IMS (*IP Multimedia Subsystem*) constitue le standard des réseaux NGN, c'est une architecture qui marque le développement d'un nouveau type de réseaux qui s'inscrit à la croisée d'IP et des télécoms. Comme son nom l'indique, IMS définit un sous-réseau multimédia ou plutôt un réseau parallèle aux réseaux actuels. Son ambition est le traitement des flux et services multimédias dans une optique de convergence, quel que soit le réseau d'accès utilisé (GSM, Wi-Fi, Ethernet, UMTS, WiMAX, etc.), quelle que soit sa nature (fixe ou mobile), et quel que soit le type de terminal considéré (ordinateur, smartphone, téléphone, etc.). L'IMS offre aux opérateurs la possibilité de construire une infrastructure de services ouverte basée sur IP avec un déploiement facile de nouveaux services, et sur le protocole SIP (*Session Initiation Protocole*) pour le contrôle de session.

Motivations et contributions

L'aspect partagé de la couche IP expose les opérateurs à des problèmes de grande envergure à savoir : la sécurité de l'information. En effet, la sécurité constitue un point crucial

dans la mesure où elle représente un des principaux moyens pour gagner la confiance des clients.

L'objectif de cette thèse est d'améliorer les mécanismes de sécurité utilisés au niveau d'IMS sans affecter la performance, en terme de temps de traitement et de nombre de messages de signalisation échangés, voire même l'améliorer, afin de proposer un environnement entièrement sécurisé et avec une expérience utilisateur assurée.

Nous nous focalisons dans un premier lieu sur la phase d'enregistrement puisqu'elle est la première action réalisée par un terminal, dès sa mise en route. Cette phase d'enregistrement est indispensable puisqu'elle permet à la fois d'accéder aux différents services proposés et d'être joignable par ses correspondants. Notre méthode d'analyse repose sur la quantification de la signalisation induite par l'authentification IMS. La quantification est effectuée à l'aide d'expérimentations réelles. Cela nous a permis également de valider le protocole que nous proposons.

Dans un seconde temps nous nous sommes intéressés à l'utilisateur final. En effet, dans un marché compétitif, les utilisateurs aiment profiter de la liberté d'utiliser les services de tout fournisseur de contenu en fonction de leurs besoins et préférences. Toutefois, jusqu'à maintenant l'IMS a été utilisée selon un modèle spécifique, le modèle que l'on appelle "*walled-garden*". Dans un modèle fermé (*walled-garden*), les services sont fournis aux utilisateurs du même opérateur afin que les utilisateurs n'aient pas à chercher des applications en dehors du réseau IMS. Ces applications sont hébergées par l'opérateur de réseau IMS, qui garde le contrôle total sur les utilisateurs. Cette façon de faire est très restrictive pour les utilisateurs, car ils n'ont pas le choix de choisir les applications qu'ils souhaitent souscrire. Ils sont en effet restreints à ce qui est offert par leur opérateur de télécommunications. Pour attirer les opérateurs et prestataires de services, IMS doit démontrer qu'il est en effet une architecture multi-service qui peut être utilisée comme un cadre de services commun, même pour les services non-SIP, et certainement au moins pour les services Web. La création de nouveaux modèles fermés n'est pas une stratégie qui est durable pour les opérateurs dans les années à venir. Compte tenu de la prolifération des services d'Internet, le succès éventuel de l'IMS serait proportionnel au trafic généré entre IMS et l'Internet. Une architecture IMS avec un très faible trafic de/vers Internet serait une architecture IMS qui n'a pas réussi à fournir une valeur ajoutée aux utilisateurs finaux, et par la suite les utilisateurs peuvent préférer contourner IMS pour accéder directement aux services sur Internet. Pour relever ces défis, nous proposons une extension du modèle IMS existant pour accéder aux applications IMS qui sont situées en dehors du domaine IMS et entretenus par d'autres opérateurs de services. Nous

avons appelé ce nouveau modèle "*virtual walled-garden model*" ou modèle muré/fermé virtuel. Ce modèle permettra de créer un lien de confiance entre le domaine IMS et les services externes, et permettra de réduire la charge des utilisateurs ainsi que les fournisseurs de services SPs (*Services Providers*) par l'utilisation d'un mécanisme SSO (*Single Sign-On*) en utilisant une fédération d'identité.

Organisation

La présentation des travaux s'organise en 5 chapitres que nous synthétisons de la façon suivante :

Chapitre 1 - Architecture IMS : Ce chapitre explicite les notions sur lesquelles s'appuie la thèse. Il présente les motivations et intérêts pour un opérateur de déployer l'IMS. Il introduit par la suite l'IMS dans sa globalité, puis développe son architecture.

Chapitre 2 - Sécurité en IMS : Ce chapitre présente les attaques potentielles, l'architecture de sécurité en IMS et les associations de sécurité, ainsi que les différents mécanismes de sécurité.

Chapitre 3 - Mécanismes d'authentification : Une synthèse non exhaustive des mécanismes d'authentification est présentée. L'objectif est de donner les prérequis nécessaires pour bien comprendre le protocole proposé dans le chapitre 4.

Chapitre 4 - The PKI-based One-way IMS-AKA : L'objectif de ce chapitre est de présenter notre protocole proposé pour l'authentification et l'accord de clés (*Authentication and key agreement*) dans le réseau IMS. Ce protocole basé sur une infrastructure PKI permet de résoudre les problèmes de sécurité dont l'IMS-AKA souffre. Il permet également d'assurer une confiance au niveau du réseau de cœur de l'IMS. Le protocole proposé (*one-way*) améliore le protocole IMS-AKA en termes de sécurité, et d'efficacité par la réduction du nombre de messages échangés entre l'utilisateur et le réseau.

Chapitre 5 - Virtual Walled-Garden Model for IMS Services Provisioning : Dans ce chapitre, nous proposons un nouveau modèle de fourniture de services, ce que nous appelons *virtual walled-garden*. L'objectif est de créer un univers autonome dans lequel les abonnés sont autorisés à profiter de tous les services et contenus offerts non seulement par leur

opérateur, mais aussi par d'autres fournisseurs de services (SP : Services Providers), dans un environnement entièrement sécurisé et avec une expérience utilisateur et une qualité du service assurées. Pour relever ces défis, nous proposons une extension du modèle IMS existant pour accéder aux applications IMS qui sont situées en dehors du domaine IMS et entretenus par d'autres fournisseurs de services. Ce modèle permettra de créer un lien de confiance entre le domaine IMS et services externes, et permettra de réduire la charge des utilisateurs finaux ainsi que les SPs par l'utilisation d'un mécanisme SSO multi niveaux (*Multi-Level Single Sign-On : ML-SSO*), réalisé par la fédération d'identités.

Chapitre 1 : Architecture IMS

Dans le paysage technologique actuel, il existe différents réseaux (fixe, mobile, sans-fil,…), standardisés par différentes organisations (ITU-T, GSM-ETSI, 3GPP, IEEE,…). Tous ces réseaux ont été développés avec des intentions et des objectifs indépendants les uns par rapport aux autres. D'ailleurs il n'y a aucun concept commun entre eux qui permettrait un point de facturation commun, l'utilisation d'une adresse IP unique, la fourniture de services communs (VoIP, vidéoconférence, messagerie instantanée, jeux multi-joueurs,…), l'utilisation d'une identification commune comme le numéro de téléphone ou l'identité de l'utilisateur, indépendamment de la technologie utilisée et du lieu d'accès, tout en garantissant une qualité de service.

C'est tout l'enjeu de la technologie IMS (*IP Multimédia Subsystem*) : réaliser la convergence des réseaux fixe et mobile, être joignable où que l'on soit, sur ordinateur comme sur mobile ou autre, tout en bénéficiant de l'étendue de l'offre multimédia.

A ce titre, nous allons dans un premier temps développer, dans ce premier chapitre, les motivations et intérêts pour un opérateur de déployer l'IMS. Dans un deuxième temps, nous clarifions les relations entre NGN et IMS, ainsi nous définissons IMS dans un troisième temps. Dans un quatrième et dernier temps nous présentons l'architecture IMS et ces principaux protocoles. Enfin nous terminerons ce chapitre par une conclusion.

1.1. Motivation pour l'architecture IMS

L'Internet supporte depuis déjà plusieurs années, et avec une qualité acceptable, de nombreux services à succès tels que le courrier électronique, le Web, le streaming audio/vidéo et le chat. Dans les domaines des applications de téléphonie et les communications multimédia, Yahoo, AOL et Skype, … sont déjà présents sur ce marché mais proposent des solutions propriétaires.

La téléphonie devient donc une application sur Internet parmi d'autres, et tout fournisseur d'applications sur Internet peut proposer le service de téléphonie sur IP à ses clients indépendamment du type d'accès à Internet utilisé par le client : ADSL, câble, UMTS….

Dans ce contexte les opérateurs de télécommunication, dont le service de téléphonie était jusqu'à présent le cœur de métier (*core business*), se trouvent face à l'alternative suivante :

- Repositionner leur métier autour des applications sur IP incluant la téléphonie, devenant ainsi opérateur de services globaux. Les opérateurs qui feront ce choix devront rapidement développer une architecture IMS seule solution normalisée dans le monde télécom, et cela avant que des solutions propriétaires ne soient trop largement adoptées.
- Abandonner le marché des applications y compris celui de la téléphonie et réduire leur business à celui de fournisseur d'accès et/ou de transporteur de paquets IP. Les opérateurs qui feront ce choix limiteront leurs champs d'action à celui d'opérateurs de réseaux. Parmi les risques de cette option, la difficulté à maintenir le revenu dans un contexte où l'accès comme le transport seront devenus des commodités sujettes à une très forte pression sur les prix.

L'IMS, normalisé par le monde des télécommunications, est une nouvelle architecture basée sur de nouveaux concepts, de nouvelles technologies, de nouveaux partenaires et un nouvel écosystème. L'IMS supporte sur un réseau tout IP les sessions applicatives temps réels (voix, vidéo, conférence,…) et non temps réel (Push To Talk, Présence, messagerie instantanée,…). L'IMS intègre de plus le concept de convergence de services supportés indifféremment par des réseaux de natures différentes : fixe, mobile ou Internet. L'IMS est également désigné sous le vocable de NGN (*Next Generation Network*) multimédia.

Déployer une architecture IMS est donc une décision stratégique qui peut être prise par un opérateur télécom traditionnel dans le cadre du repositionnement de son activité sur le marché des services sur IP ; ce choix peut par ailleurs être pris par toute entité qui déciderait, même sans posséder de réseaux d'accès ou de transport, de développer une activité de services à valeur ajoutée sur IP.

L'acquisition des fondements architecturaux et normatifs de l'IMS, en particulier les spécifications de protocoles et interfaces spécifiques tels que SIP (*Session Initiation Protocol*), Diameter, COPS,.., ainsi que la connaissance des solutions déjà disponibles sur le marché, s'avèrent donc essentielles pour tout acteur/opérateur de réseaux ou de services, fournisseur d'équipements, ou clients qui souhaite prendre sa place dans le marché émergeant des services sur IP.

1.2. Relations entre NGN et IMS

Pour pouvoir comprendre l'IMS, qui reste encore jeune et en phase d'évolution, il faut avant tout bien assimiler le concept NGN qui présente un cadre plus mature pour les nouvelles

générations et une phase initiale pour le déploiement de l'IMS. De ce fait, nous essayons dans cette section d'étudier le principe de base du NGN ainsi que les différents changements et évolutions qui ont abouti à l'introduction de l'IMS et ceci en termes de principes, architecture, entités et protocoles.

1.2.1. Définition des NGN

L'acronyme NGN est un terme générique qui englobe différentes technologies visant à mettre en place un concept, celui d'un réseau convergent multiservices. Cependant, il n'existe pas encore une définition unique de la notion de NGN.

L'ITU-T a publié plusieurs recommandations concernant les réseaux NGN (série Y.2000-Y.2999). La première recommandation, Y.2001, définit les principales caractéristiques d'un réseau NGN, tandis que la recommandation Y.2011 par exemple, propose une architecture fonctionnelle. Cette architecture est repartie du fait qu'il faut séparer organiquement les trois plans : service, contrôle et transport.

Le comité technique TISPAN (*Telecoms & Internet converged Services & Protocols for Advanced Networks*) de l'ETSI (*EuropeanTelecommunications Standards Institute*) a également défini une architecture fonctionnelle pour les réseaux NGN largement inspirée de celle proposée par l'ITU-T. Les définitions des organismes de normalisation tels que l'ETSI et l'ITU-T restent toutefois assez vagues et dressent une liste générale des principales caractéristiques des réseaux NGN qui se veulent multi-réseaux, multiservices, multi-protocoles et multi-terminaux.

L'ETSI par exemple, définit le NGN comme étant « *un concept pour définir et déployer les réseaux, qui du fait de leur* ***séparation formelle en différentes couche****s et plans et de l'utilisation d'****interfaces ouvertes****, offrent aux fournisseurs de services ainsi qu'aux opérateurs une plateforme évolutive pour créer, déployer et gérer des services multimédias innovants* ».

1.2.2. Architecture des réseaux NGN

Afin de répondre aux différentes exigences citées précédemment et d'intégrer ainsi les infrastructures de télécommunication existantes avec celles à venir au sein d'une seule et

unique infrastructure commune, flexible et évolutive, des impératifs ont été clairement identifiés par les organismes œuvrant pour les réseaux NGN (3GPP, ITU-T, ETSI,…) :

- Un cœur de réseau unique et mutualisé pour tous les types de réseaux d'accès et de services,
- Une architecture de cœur de réseau en deux couches : Transport et Services,
- Une évolution du transfert des données vers le mode paquet,
- Des interfaces ouvertes et standardisées entre chaque couche afin de réaliser l'indépendance des services vis-à-vis du réseau,
- Un découplage entre la fourniture de service et la fourniture de réseau.
- Le support de technologies d'accès multiples,
- Le support de la convergence des réseaux voix/données et fixe/mobile,
- Le support de terminaux multiples (modulaires, multimode, multimédia et adaptatifs).

Ainsi, la principale caractéristique d'un réseau de nouvelle génération est son fondement sur IP qui offre un mode de transfert homogène de bout en bout, indépendant d'une part, des réseaux sous-jacents et, d'autre part, du type de données applicatives véhiculées. Après l'évolution du cœur de réseau vers le "tout-IP", la notion la plus importante reste la décomposition en plans fonctionnels séparés par des interfaces ouvertes qui assure à la fois le passage à l'échelle et la flexibilité d'une telle architecture en offrant une facilité d'interconnexion et d'intégration de nouveaux services [KnMT05, YBDC06].

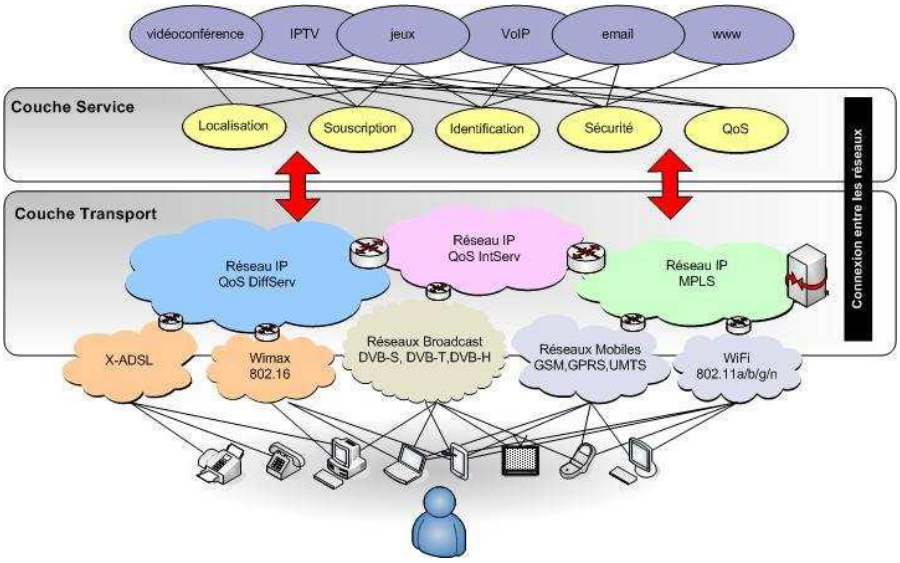


Figure 1.1. Offre une représentation communément admise de l'architecture des réseaux NGN

Le plan de "Transport" regroupe l'ensemble des ressources mises en place pour assurer le transfert de données. Il gère ainsi l'acheminement du trafic vers sa destination en fournissant une connectivité IP aux différents composants d'un réseau NGN tout en garantissant une QoS de bout en bout. Ce plan dépend directement de la technologie du réseau de transport utilise pour acheminer les paquets.

Le plan de "Service" fournit les fonctionnalités de base pour l'exécution des services avec ou sans session. En effet, il regroupe les plateformes d'exécution de service et de diffusion de contenu tout en masquant la diversité technologique aux clients et aux fournisseurs de services.

Après ces plans horizontaux, on peut différencier les réseaux d'accès des réseaux de transit. En effet, les premiers ont pour but de concentrer le trafic des différents utilisateurs vers des équipements centraux. Alors que, les deuxièmes ont pour vocation d'acheminer des volumes de trafic importants entre quelques entités limitées. Dans la figure 1.1, nous pouvons noter la diversité des technologies permettant à l'utilisateur d'accéder aux services, ce qui peut former ce qu'on appelle parfois le plan d'accès. Il est important de noter que l'interaction entre les réseaux de nouvelle génération est les réseaux traditionnels est prise en charge par différentes passerelles d'interconnexion afin d'assurer une compatibilité avec les technologies déployées actuellement.

1.2.3. Types de NGN

Il existe trois types de réseau NGN : NGN Classe 4, NGN Classe 5 et NGN Multimédia. Les NGN Classe 4 et Classe 5 sont des architectures de réseau offrant uniquement les services de téléphonie. Il s'agit donc de NGN téléphonie. Dans le réseau téléphonique commuté (RTC), un commutateur Classe 4 est un centre de transit. Un commutateur Classe 5 est un commutateur d'accès aussi appelé centre à autonomie d'acheminement. Le NGN Classe 4 (respectivement NGN Classe 5) émule donc le réseau téléphonique au niveau transit (respectivement au niveau accès) en transportant la voix sur un mode paquet. Quant au NGN Multimédia, c'est une architecture offrant les services multimédia (messagerie vocale/vidéo, conférence audio/vidéo, tonalité ou sonnerie de retour (Ring-back tone) voix/vidéo) puisque l'usager a un terminal IP multimédia. Cette solution est plus intéressante que les précédentes puisqu'elle permet à l'opérateur d'innover en termes de services par rapport à une solution NGN téléphonie qui se cantonne à offrir des services de téléphonie.

Pour résumer, le NGN Classe 4 permet :

- Le remplacement des centres de transit téléphoniques (Classe 4).
- La croissance du trafic téléphonique en transit.

Le NGN Classe 5 permet :

- Le remplacement des centres téléphoniques d'accès (Classe 5).
- La croissance du trafic téléphonique à l'accès.
- La voix sur DSL/Voix sur le câble.

Le NGN Multimédia permet d'offrir des services multimédia à des usagers disposant d'un accès large bande tel que xDSL, câble, WiFi/WiMax, EDGE/UMTS, etc. Le Multimédia NGN est parfois appelé IMS.

1.3. IMS : IP Multimedia Subsystem

L'introduction de l'IMS dans les réseaux fixe et mobile représente un changement fondamental dans les réseaux de télécommunication de type voix. Les nouvelles capacités des réseaux et des terminaux, l'association entre l'Internet et la voix, le contenu et la mobilité donnent naissance à de nouveaux modèles de réseaux, et surtout offrent un formidable potentiel pour développer de nouveaux services. Dans cet objectif, l'IMS est conçu pour offrir aux utilisateurs la possibilité d'établir des sessions multimédia en utilisant tout accès haut débit et une commutation de paquets IP.

1.3.1. Définition

L'IMS a été initialement défini par le 3GPP (*3rd Generation Partnership Project*) [Gpp00], qui est une collaboration entre un certains nombres d'organismes de normalisation dans le secteur des télécommunications (ETSI pour l'Europe).
IMS a été introduit dans la 3GPP release 5 (2003), dans laquelle SIP (*Session Initiation Protocol*), défini par l'IETF (*Internet Engineering Task Force*), a été choisi comme protocole principal pour l'IMS. Ont suivi d'autres releases de 3GPP, qui ont ajouté des fonctions additionnelles, comme la gestion de présence et de groupe, l'interconnexion avec les WLAN, WiMax, … L'organisme planche actuellement sur la version 12.
Un des groupes de travail de l'ETSI, TISPAN (*Telecommunications and Internet converged Services and Protocols for Advanced Networking*) [Etsi00] a normalisé IMS comme NGN. Si

3GPP est plus axé sur le point de vue opérateur mobile, TISPAN ajoute des spécifications plus du point de vue filaire, pour étendre l'intégration des réseaux fixes, afin de permettre la convergence fixe/mobile.

Un autre organisme, le 3GPP2, a aussi normalisé son propre IMS. 3GPP2 est né sous l'impulsion d'organismes Nord-Américains et Asiatiques, et oppose au duo GSM/UMTS, les technologies CDMA/CDMA2000.

La première release de 3GPP2 sur l'IMS reprend largement la release 5 de 3GPP. Les deux réseaux IMS définis par les deux organisations sont en réalité très similaires, mais sont marqués par quelques différences. 3GPP2 l'a complété par quelques ajustements relatifs à leurs préoccupations. Quoi qu'il en soit, le but de ces deux organismes est d'assurer, aux applications IMS, leur fonctionnement à travers les différentes infrastructures réseaux.

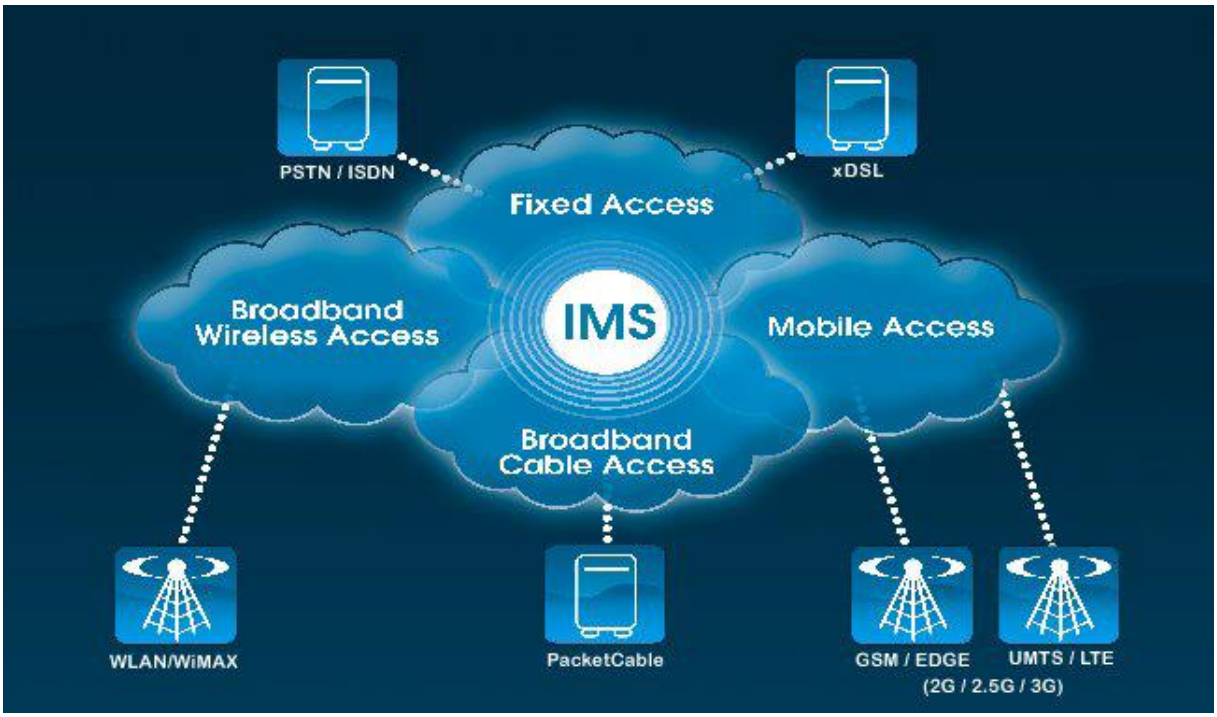


Figure 1.2. Vue simplifiée d'IMS par TISPAN [Etsi11]

La définition d'IMS est très dépendante de l'entité qui le considère. Pour l'ETSI, via notamment TISPAN, il s'agit d'offrir une convergence entre les technologies d'accès comme la figure 1.2 l'illustre. Sur cette figure apparaît clairement l'objectif principal de l'architecture IMS telle qu'envisagée par TISPAN : l'intégration de différents supports physiques. On retrouve ici les différents média physiques du 3GPP (2G, 2.5G et 3G) mais aussi les réseaux d'accès filaires de type xDSL, RTC, RNIS et câble. Enfin, les accès sans-fil sont aussi

intégrés avec les WLAN et le WiMAX. IMS est alors considérée ici comme une architecture de convergence des différents réseaux d'accès, sans fil et filaire, avec les réseaux de mobiles. Pour un constructeur comme CISCO, la présentation IMS est différente (figure 1.3). Ainsi, on peut observer un autre point clef de l'architecture IMS, au-delà de la convergence des supports, c'est l'intégration des différents services qui est visée en offrant une plateforme commune, facilitant leur déploiement et centralisant des fonctionnalités comme la gestion de session, les fonctions d'authentification, autorisation, et traçabilité connues sous le sigle AAA (*Authentication, Authorization, Accounting/Auditing*), ou encore la gestion de la QoS.

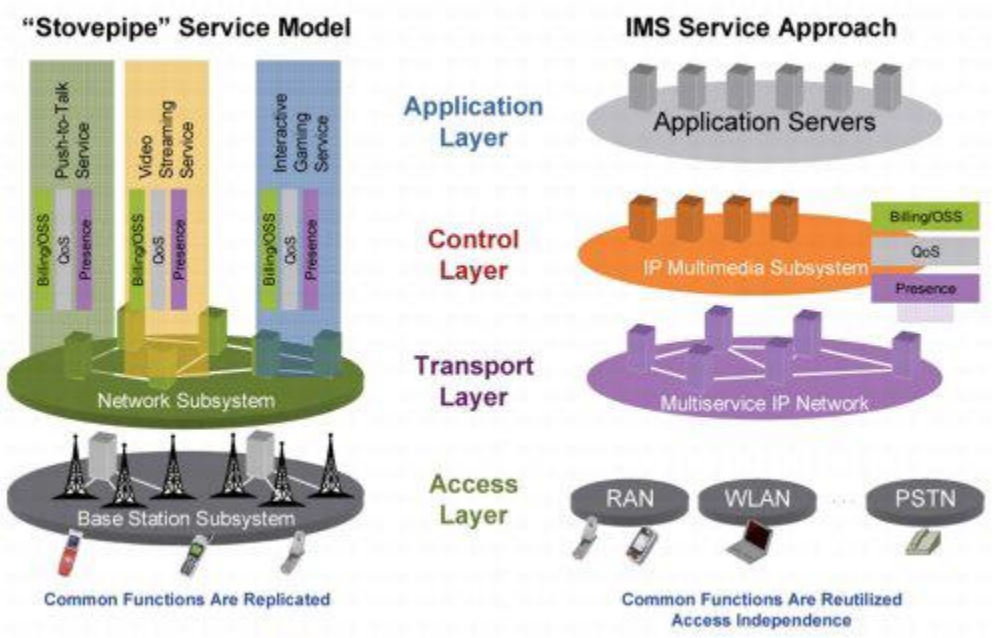


Figure 1.3. Vue simplifiée d'IMS par CISCO

Par conséquent, on peut dire que IMS est une déclinaison plus réduite et plus précise des NGN. L'objectif reste le même : définir une architecture distribuée indépendante de l'accès client, qu'il soit sur réseaux fixe, mobile, internet. En outre, les réseaux IMS seront à ce titre axés sur les services, et fourniront tous les moyens nécessaires pour en offrir de nouveaux, mais aussi pour adapter les existants pour augmenter les recettes [Bert07, Gonz00].

1.3.2. Bénéfices de l'architecture IMS

Etre accessible où que l'on soit, naviguer sur Internet, consulter ses mails, tout cela est déjà possible avec le réseau cellulaire 2.5G ou 3G. Qu'apporte donc de plus l'IMS ?

1.3.2.1. Une plateforme commune

L'IMS, c'est avant tout une plateforme commune à tout service. En effet, un des buts principaux de l'IMS est de rendre la gestion du réseau plus facile. Pour ce faire, IMS est situé entre les couches de transport et les couches applicatives. IMS agit donc comme une couche totalement indépendante, quels que soient les terminaux et équipements en jeu, les protocoles utilisés,… séparant ainsi le réseau de transport de base et les applications. De plus, IMS est également découpée en différentes couches. Mais on peut déjà noter que cette plateforme commune permettra de réduire le temps de mise en œuvre d'un nouveau service. En effet, au vue de l'architecture actuelle, ou plutôt des architectures actuelles, dès qu'un opérateur souhaite proposer un nouveau service, il doit penser au préalable à aménager son réseau. Cela passe par les équipements, les liens, les connexions,… Chaque technologie actuelle (GPRS, WLAN, fixe,...) possède ainsi son propre réseau cœur. Ce qui est lourd et onéreux comme investissement et mise en exploitation pour un opérateur. Avec l'IMS, tout nouveau service viendra se greffer à l'architecture standardisée déjà en place.

1.3.2.2. La qualité de service

Un deuxième atout est l'émergence de la Qualité de Service (QoS). Actuellement, malgré l'augmentation de la bande passante et de l'accessibilité au réseau Internet avec la 3G, il n'y a malheureusement aucune garantie quant à la qualité de service. La 3G fonctionne au "best effort", c'est-à-dire que le maximum sera fait pour bénéficier d'une bande passante adéquate, mais rien n'indique ni contrôle l'adéquation de cela à nos besoins de QoS. En effet, le réseau n'offre pas une garantie concernant la quantité de bande passante qu'un utilisateur obtient pour une connexion particulière ou sur le délai de bout en bout des paquets. Par conséquent, la qualité d'une conversation VoIP, par exemple, peut varier considérablement tout au long de sa durée. Pour résoudre ce problème, les mécanismes de QoS ont été développés dans le monde IP pour assurer une garantie de bande passante pendant la durée de transmission. L'IMS bénéficiant du réseau IP, ces mécanismes vont donc pouvoir être mis en œuvre ; ainsi IMS peut fournir la qualité de service requise pour profiter des sessions multimédia en temps réel dans les meilleures conditions.

1.3.2.3. La facturation

Un troisième point essentiel pour un opérateur concerne le domaine de la facturation. Avec la 3G actuelle, si un utilisateur utilise le service de vidéoconférence, il y a un volume

important de données transférées, résultant de la transmission simultanée d'audio et vidéo. Cela revient donc très cher à l'utilisateur, puisque l'opérateur a pour habitude de facturer en fonction du nombre d'octets transférés. Si l'opérateur choisissait un autre mode de facturation, indépendamment du volume d'informations traitées, ce serait bénéfique pour l'utilisateur, mais cela pourrait être disproportionné par rapport à la charge réseau occasionnée.

L'avantage de l'IMS est qu'il fournit des informations sur le type de service demandé par l'utilisateur, permettant ainsi aux opérateurs de déterminer la façon de facturer en fonction du service consommé.

1.3.2.4. Une architecture IP de bout en bout

Le dernier point est qu'IMS est basé sur une architecture IP de bout en bout. Un problème majeur avec la technologie cellulaire concerne le *roaming* en fonction du service en cours d'utilisation.

Avec IMS, tout est traité en IP. Ainsi, l'utilisation des protocoles du monde Internet permet aux utilisateurs d'être mobiles, indépendamment du pays dans lequel ils se trouvent, tout en étant capable d'utiliser tous les services disponibles, comme s'ils étaient dans leur réseau mère (*home network*).

1.4. Architecture IMS

1.4.1. Architecture fonctionnelle en couche d'IMS

Pour une gestion facilitée, IMS s'appuie sur une plateforme commune à tout service. Ainsi, on peut segmenter le réseau cœur IMS en quatre parties distinctes :

- la couche physique (accès)
- la couche transport
- la couche de session (contrôle)
- la couche application.

Chaque couche est indépendante, de sorte qu'il est possible, par exemple, d'ajouter librement de nouveaux services dans la couche applicative, sans tenir compte du réseau d'accès que les utilisateurs ont employé, ni du terminal qu'ils ont utilisé. Cela dit, il est souhaitable que les serveurs d'applications adaptent leur réponse en fonction de ces critères : par exemple, une page Web ne doit pas contenir les mêmes informations ni être formatée de la même manière selon qu'elle est destinée à être visualisée sur un ordinateur portable ou sur un téléphone

portable [QKAW09]. Nous allons voir ces différentes couches, en partant de la plus basse à la couche la plus haute (figure 1.4).

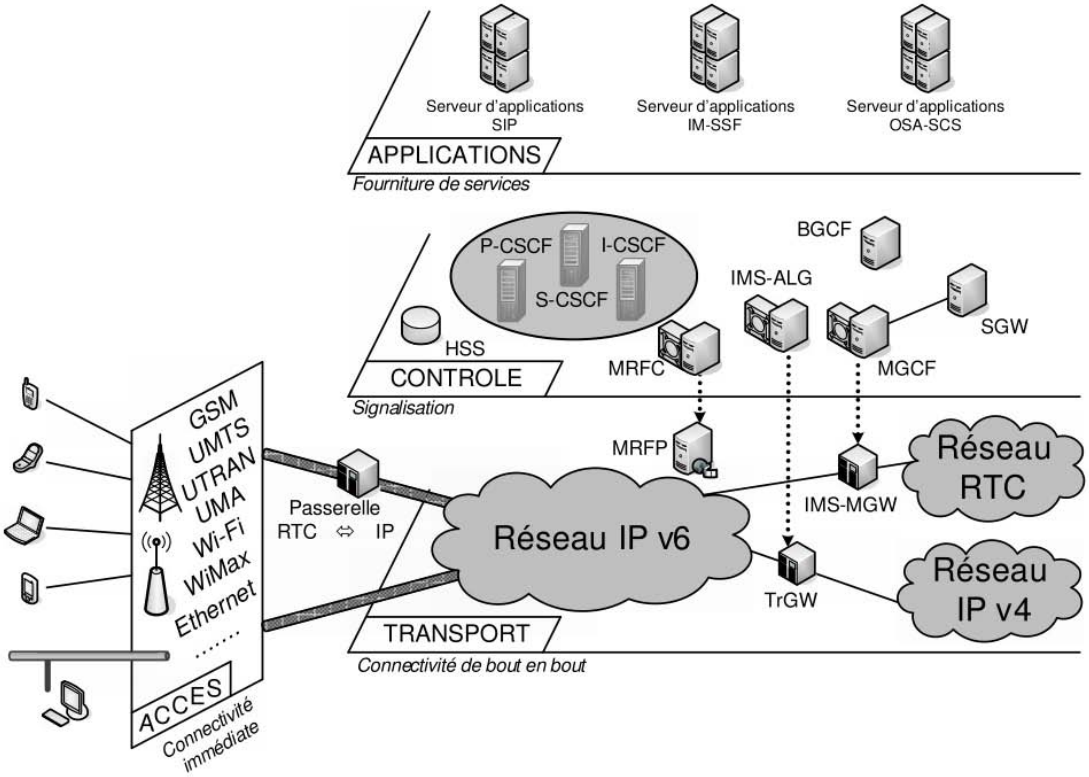


Figure 1.4. Architecture de l'IMS

1.4.1.1. La couche physique (accès)

La couche physique est une couche d'accès à l'architecture IMS. Elle définit la manière dont l'utilisateur se connecte au réseau. IMS permet un large choix de terminaux aux utilisateurs. En effet, tous les systèmes comme les ordinateurs, les téléphones mobiles, les PDAs, les téléphones fixes numériques, sont capables de se connecter à l'architecture IMS via le réseau. Les téléphones traditionnels analogiques pourraient également se connecter à l'architecture IMS par le biais de passerelles, pour ainsi obtenir une adresse IP, condition nécessaire à une connexion au réseau cœur IMS.

En regroupant différents types de réseaux d'accès au sein de cette couche, IMS offre un niveau d'abstraction à la manière dont l'utilisateur se connecte au réseau. Cette vision est à l'origine de l'idée de convergence des réseaux vers un réseau unique, prônée par l'IMS. Autrement dit, IMS est indépendant du réseau d'accès, qui n'est qu'un élément assurant la connectivité de l'utilisateur au réseau cœur.

1.4.1.2. La couche transport

Cette couche permet la connectivité de bout en bout entre les différents interlocuteurs. Alors que la couche d'accès se contente de connecter un utilisateur au réseau IMS, la couche transport se charge de l'acheminement des données de l'utilisateur jusqu'à son (ou ses) correspondant(s). Cela comprend le transport de l'information par les routeurs et le choix de la route empruntée dans le réseau. C'est le réseau IP qui est utilisé dans cette couche. Un des principaux enjeux au niveau de cette couche est de réaliser la convergence entre réseau à routage par paquet et réseaux à routage par circuit ; autrement dit, entre le réseau téléphonique commuté et le réseau utilisant TCP/IP. On pense ici aux téléphones traditionnels analogiques. C'est à ce niveau donc, qu'interviennent les passerelles PSTN (*Public Switched Telephone Network*).

1.4.1.3. La couche contrôle de session (contrôle)

Cette couche assure la gestion et le contrôle du réseau. Elle est en charge de tous les messages de signalisation dans le réseau, permettant d'ouvrir, de maintenir, de modifier et de terminer une session entre des utilisateurs. C'est la partie intelligente du modèle, qui offre toutes les fonctionnalités de gestion des utilisateurs et constitue le véritable socle de l'IMS. Nous la détaillons plus loin, en présentant l'ensemble des entités qui s'inscrivent dans cette architecture.

1.4.1.4. La couche application

Elle consiste en la fourniture des services, qu'ils soient audio, vidéo ou textuels. Cette couche implémente tous les services que l'on peut proposer aux utilisateurs. Elle est la partie la plus ouverte du modèle, puisque le réseau IMS ne spécifie pas les services eux-mêmes, mais offre une plate-forme de déploiement unifiée, simple, rapide, productive et sécurisée pour la mise en place de nouveaux services.

Tout service est exécuté par un serveur applicatif, en liaison avec les équipements de la couche de session par l'intermédiaire des protocoles SIP et Diameter, assurant ainsi la sécurité des utilisations. A noter qu'un serveur peut exécuter différents services comme de la téléphonie et un service de messagerie. L'avantage de cette flexibilité est de réduire la charge de travail au niveau de la couche de contrôle.

1.4.2. Principaux composants de l'architecture IMS

Dans cette partie, on va faire une description synthétique des différents composants de l'architecture IMS. Toutes ces entités constituent le socle de l'architecture IMS et, à ce titre, sont indispensables pour le fonctionnement d'un réseau IMS. Elles sont des entités logiques, ce qui signifie que, malgré leur distinction fonctionnelle, rien n'empêche de les implémenter (toutes ou certaines) au sein d'un même équipement.

1.4.2.1. Le Home Subscriber Server (HSS)

Le HSS est la base de données des abonnés de l'IMS et des services associés (à l'instar du HLR pour les réseaux mobiles). Les comptes utilisateurs, leurs profils, droits d'accès, et nom du S-CSCF (*Serving-Call Session Control Function*) associés sont stockés dans cet équipement. Il communique avec les CSCFs pour fournir, temporairement, une copie du profil utilisateur.

Toutes les données relatives à un même compte utilisateur doivent être stockées sur un même HSS. Néanmoins, lorsque le nombre d'utilisateurs est important, il est possible de les repartir au sein de plusieurs HSS. Dans ce cas, il est nécessaire de mettre en place une entité complémentaire, appelée SLF (*Subscription Locator Function*), qui a pour rôle de déterminer le HSS contenant les données relatives à un utilisateur.

Dans la suite, nous nous contenterons de parler d'un serveur HSS, même s'il peut y en avoir plusieurs, auquel cas, il s'agirait d'un serveur SLF qui relaie les requêtes vers le HSS concerné.

Les serveurs HSS et SLF communiquent avec les autres entités du réseau au moyen du protocole Diameter.

1.4.2.2. CSCF: Call Session Control Function

Les CSCF sont des serveurs de contrôle de sessions, qui utilisent le protocole de signalisation SIP pour communiquer. Ils sont les organes de traitement des références dans l'architecture IMS et réalisent les fonctions permettant d'orienter et de contrôler une session.

Concrètement, lorsqu'un utilisateur se connecte, il peut accéder à un réseau qui n'est pas celui de l'opérateur chez qui il a souscrit un service. Il faut donc que l'utilisateur emprunte le réseau pour assurer sa connectivité, tout en cherchant à se relier à des entités appartenant à son opérateur, pour avoir accès aux services de son contrat.

Il existe trois types de CSCF, Proxy CSCF, Interrogating CSCF et Serving CSCF, chacun de ces serveurs peut se trouver en nombre dans un réseau IMS, notamment pour répondre à la charge des demandes. Nous détaillons dans les sections qui suivent le rôle de chacun de ces serveurs.

1.4.2.2.1. Le Proxy CSCF

Le *Proxy CSCF* est toujours le premier point de contact entre un terminal et le réseau IMS. Ses missions consistent notamment à contrôler l'accès et à établir une connexion sécurisée avec le terminal. Son adresse est découverte par l'utilisateur lors d'une phase de "*CSCF discovery*". Il agit comme intermédiaire entre l'abonné et l'I-CSCF.

1.4.2.2.2. L'Interrogating CSCF

L'*Interrogating CSCF* a comme principales fonctions de déterminer le S-CSCF auquel l'abonné peut se connecter et transmettre les messages entre le P-CSCF et le S-CSCF, un peu comme une passerelle.

1.4.2.2.3. Le Serving CSCF

Le *Serving CSCF* est l'équipement qui a pour rôle de finaliser l'authentification de l'utilisateur et lui procurer les services opérationnels. Il fournit des informations de routage, de facturation, maintient l'état de la session en contrôlant un compteur de temps (*timer*), interroge le HSS pour vérifier les droits utilisateurs vis-à-vis d'un service, etc. En résumé, le S-CSCF est le cerveau du réseau cœur IMS. Pour arriver à ces fins, le S-CSCF assure notamment :

- La gestion des enregistrements SIP des utilisateurs, liant l'emplacement courant de l'utilisateur (URI privé) et à son identité (URI public).
- Le contrôle de toute signalisation SIP provenant et à destination de son client.
- La sélection des serveurs d'applications.

Le serveur S-CSCF est déterminé grâce au serveur I-CSCF. Il en informe alors le P-CSCF, de manière que ce dernier puisse ultérieurement s'adresser directement au S-CSCF. Parallèlement, le S-CSCF enregistre dans le HSS la position de l'abonné dans le réseau et indique au HSS son adresse, afin qu'une entité cherchant à joindre l'abonné détermine le S-CSCF auquel elle doit s'adresser.

Une fois désigné pour servir la session d'un utilisateur, le premier rôle du S-CSCF est de récupérer auprès de la base HSS via le protocole Diameter, l'ensemble des paramètres des profils d'utilisateur pour l'enregistrer, l'authentifier et vérifier ses droits d'accès.

De plus, il met à jour, dans cette même base HSS, l'état courant des sessions dont il a la charge, pour localiser l'utilisateur dans ses déplacements, modifier ses préférences, mais aussi avoir l'historique de ses communications utile pour la facturation.

Remarquons que le terminal de l'utilisateur lui-même ne connaît jamais l'adresse du I-CSCF, ni celle du S-CSCF. Ces informations sont masquées et ne sont connues que du P-CSCF qui sollicite les serveurs concernés pour le terminal.

Le S-CSCF est responsable des traitements à réaliser pour l'utilisateur. Il est de plus l'entité chargée d'effectuer le routage des appels. En particulier, si l'adresse de destination n'est pas une adresse SIP, mais, par exemple, un numéro de téléphone standard, le serveur S-CSCF fournit les fonctionnalités de conversion pour joindre les passerelles téléphoniques.

Enfin, le S-CSCF est chargé d'assurer l'interaction entre le client et les serveurs d'applications en commutant ses requêtes vers les serveurs adéquats.

1.4.2.3. Serveurs d'applications

Les ASs (Application Servers) sont des entités SIP fournissant différents types de services aux utilisateurs. Ils sont connectés au serveur S-CSCF, qui joue l'intermédiaire entre l'utilisateur et les services.

On distingue trois grandes familles de serveurs d'applications : SIP AS (SIP Application Server), IM-SSF (IP Multimedia-Service Switching Function) et OSA-SCS (Open Service Access-Service Capability Server).

1.4.2.3.1. SIP AS

Les serveurs SIP AS (SIP Application Server) permettent l'exécution des services nativement implémentés pour fonctionner avec SIP. Les services les plus classiques sont généralement implémentés au sein de ces serveurs.

1.4.2.3.2. IM-SSF

Pour permettre la mobilité de l'abonné tout en lui garantissant la fourniture de ses services, même s'il se trouve dans une infrastructure qui n'appartient pas à son opérateur de services, il est nécessaire d'avoir une passerelle, appelée IM-SSF (IP Multimedia-Service Switching Function), afin de connecter l'abonné au serveur d'applications de son opérateur.

IM-SSF permet ainsi d'accéder à des services distants en réalisant l'interface entre, d'un côté, le serveur S-CSCF communiquant avec le protocole SIP et, de l'autre côté, des serveurs distants en communiquant avec le protocole CAP.

1.4.2.3.3. OSA-SCS

Les serveurs OSA-SCS (Open Service Access-Service Capability Server) fournissent le moyen d'interagir avec les serveurs d'applications OSA. OSA a été défini par le 3GPP et l'ETSI comme une architecture de gestion des services dans un réseau téléphonique de troisième génération. Son objectif est de proposer une vision très abstraite du réseau, n'imposant aucune architecture et s'adaptant facilement, quelle que soit l'architecture considérée.

Pour cela, OSA fournit une API qui facilite le développement des services. Cette API, ouverte, librement téléchargeable et indépendante des technologies utilisées, a été nommée OSA Parlay. Le 3GPP y fait référence, de manière plus neutre, sous le nom d'API OSA. Ces applications sont accessibles à l'IMS grâce au serveur OSA-SCS.

1.4.3. Les interfaces

IMS décompose l'infrastructure de réseau en des fonctions distinctes avec des interfaces standardisées. Chaque interface est spécifiée comme un "point de référence", qui définit à la fois le protocole et les fonctions opérées sur chaque interface (Figure 1.5).

Cette section explique comment les entités du réseau décrites précédemment sont reliées les unes aux autres ainsi que le protocole utilisé au niveau de chaque interface.

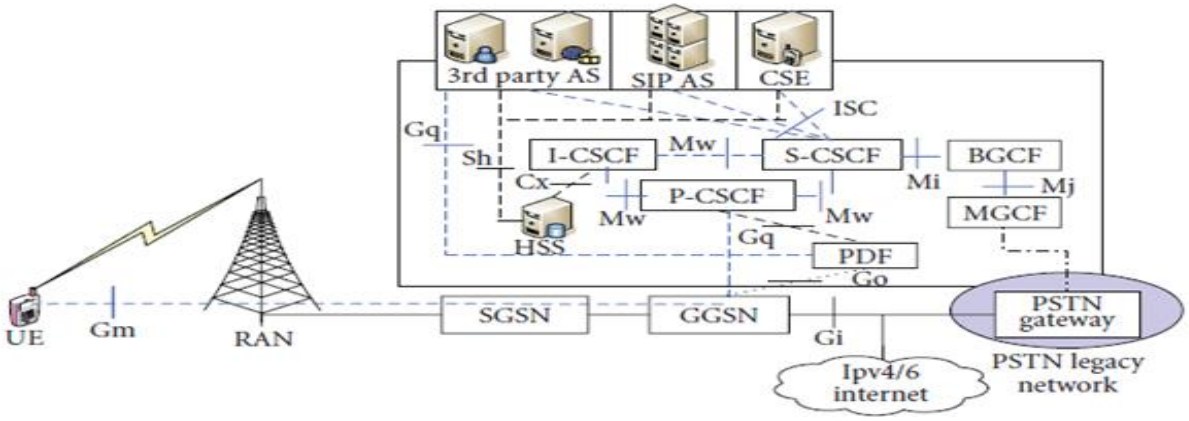


Figure 1.5. Points de référence en infrastructure IMS

1.4.3.1. Point de référence Gm :

Le point de référence Gm relie l'UE à l'IMS. Il est utilisé pour le transport de tous les messages SIP de signalisation entre l'UE et l'IMS. Les procédures dans le point de référence Gm peuvent être divisées en trois grandes catégories: l'enregistrement, la session de contrôle et les transactions.

1.4.3.2. Point de référence Mw

Mw est un point de référence basé sur SIP entre les différents CSCF. Les procédures décrites dans ce point de référence peuvent être divisées en trois catégories principales : l'enregistrement, le contrôle de session et les transactions.

1.4.3.3. Point de référence ISC:

ISC (IMS Service Control) est un point de référence pour l'envoi et la réception de messages SIP entre S-CSCF et le serveur d'application. Les procédures ISC peuvent être divisées en deux principales catégories de routage : les demandes SIP vers un AS et les requêtes SIP initiées par AS.

1.4.3.4. Point de référence Cx :

Les données centralisées dans HSS doivent être utilisés par l'I-CSCF et S-CSCF lorsque l'utilisateur reçoit des sessions. Par conséquent, il doit y avoir un point de référence entre le HSS et le CSCF. Ce point de référence est appelé Cx et le protocole choisi est le diamètre [Gpp07, Gpp10].

1.4.3.5. Point de référence Dx

Lorsque plusieurs HSS sont déployées séparément dans un réseau, aucune des deux entités suivantes l'I-CSCF et le S-CSCF ne sait quel HSS doit contacter. Dans ce cas, il est nécessaire de contacter le SLF en premier. A cet effet, le point de référence Dx a été introduit et utilisé en conjonction avec le point de référence Cx. Le protocole utilisé dans ce point de référence est basé sur le protocole Diameter [Gpp07, Gpp10].

1.4.4. Gestion des identités

Comme dans tout type de réseau, il est impératif de pouvoir identifier les utilisateurs d'une façon unique et faire en sorte qu'ils soient joignables de n'importe quel réseau. Dans IMS, il y a un nouveau concept d'identification par rapport à ce qui se faisait dans les réseaux mobile tout en restant compatible avec. Cette identification peut paraître un peu étrange et compliqué mais elle fournit plus de flexibilité pour réaliser des nouveaux services. La technique d'identification est prise du protocole SIP (*Session Initiation Protocol*).

1.4.4.1. Public User Identity

PUI est une adresse publique qui permet d'identifier un utilisateur. L'opérateur attribut une ou plusieurs adresses publiques pour chaque utilisateur IMS. De cette manière, cette nouveauté permet à l'utilisateur de séparer son identité personnelle, familiale et d'affaire pour générer des services différents. L'identité publique de l'utilisateur est l'équivalent du MSISDN (*Mobile Station ISDN Number*) en GSM, c'est donc une adresse de contacte qui permet de joindre un abonné et à router les messages SIP. La *Public User Identity* peut être sous deux formats :

- *SIP URI* : sous la forme "sip : premier.dernier@opérateur.com". Il est aussi possible d'inclure un numéro de téléphone dans une SIP URI qui sera sous le format : "sip :+33-961-007-007@opérateur.com; user = phone".
- *TEL URL* : permet de représenter un numéro de téléphone dans un format international "tel : +33-961-007-007". Il est impossible de s'enregistrer avec un TEL URL, il faut toujours une SIP URI pour se faire. Cela dit, le TEL URL est utilisé pour faire des appels entre le monde RTC et le monde IMS. En effet, en RTC, les téléphones sont identifiés par des numéros et ne peuvent composer que des numéros. L'opérateur IMS doit ainsi allouer à chaque utilisateur au moins une SIP URI et un TEL URL [MeFa04].

1.4.4.2. Private User Identity

Chaque utilisateur se voit attribuer une identité privée pour chaque utilisateur. Cette identité joue le même rôle que l'IMSI (*International Mobile Subscriber Identity*) en GSM ; elle permet en effet l'authentification et l'enregistrement de l'abonné. La PUI est en principe stockée dans la carte à puce et prend le format d'un "*Network Access Identifier*" : "username@opérateur.com".

1.4.4.3. Relations entre Public et Private User Identity

Dans le cas GSM/UMTS, la carte à puce stocke l'identité privée et au moins une identité publique. Le HSS contient pour chaque utilisateur son identité privée et la collection d'identités publiques qui lui sont attribuées. Notons que dans le cas où l'utilisateur utilise une carte GSM/UMTS qui ne contient pas ces informations, le terminal est capable de les construire à travers l'IMSI. La relation entre l'utilisateur IMS et ces identités dans la Release 5 est montré par la figure 1.6 :

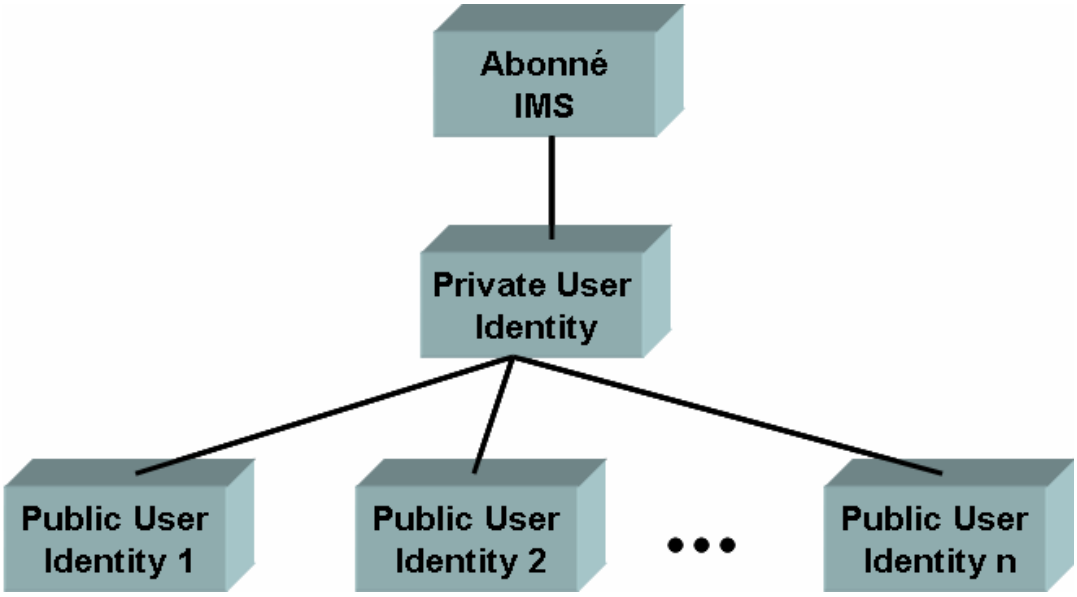


Figure 1.6. Relation entre l'identité privée et publiques en IMS 3GPP R5.

Dans l'IMS 3GPP Release 6, un abonné peut avoir plusieurs identités privées comme illustré dans la figure 1.7. Dans le cas de l'UMTS, une seule identité privée peut être contenu dans la carte à puce, même si l'utilisateur peut avoir plusieurs cartes contenant chacune une identité privée différente. Il est encore possible d'utiliser simultanément la même identité

publique avec plusieurs identités privées (deux cartes insérées dans deux terminaux différents).

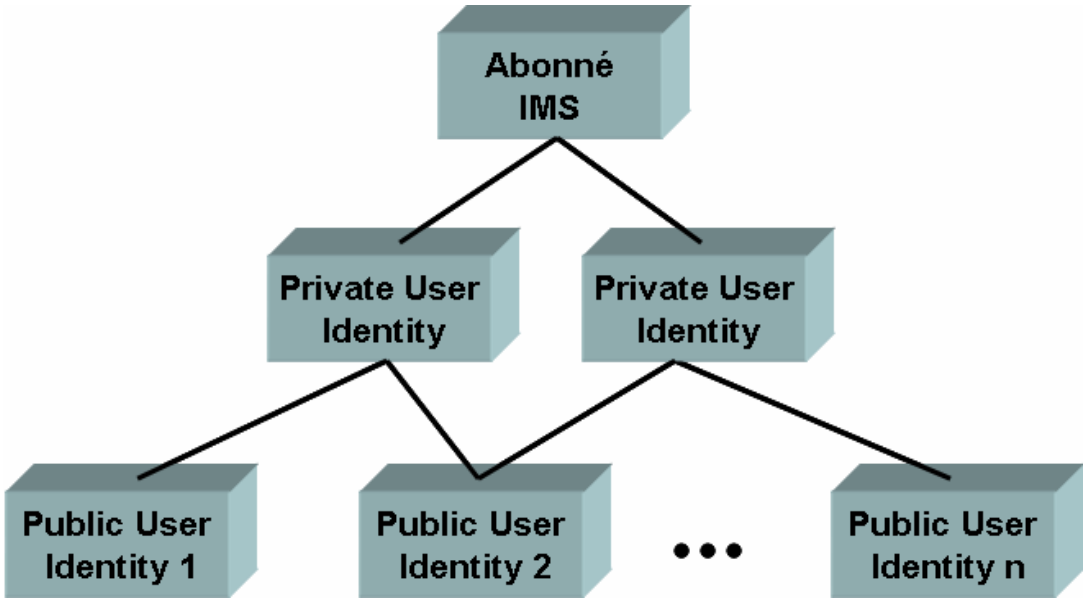


Figure 1.7. Relation entre l'identité privée et publiques en IMS 3GPP R6.

1.4.5. Carte USIM et ISIM

Dans chaque terminal, il y a une carte à puce appelée UICC (*Universal Integrated Circuit Card*). L'UICC est utilisé pour stocker des informations telles que l'état d'enregistrement, les clefs d'authentifications, les messages et un carnet d'adresses. L'UICC contient plusieurs applications logiques qui peuvent être : la SIM, l'USIM et l'ISIM.

1.4.5.1. USIM (Universal Subscriber Identity Module)

USIM est utilisée pour l'accès au réseau UMTS en mode circuit ou paquet. Elle contient les paramètres suivants :

- IMSI : comme en GSM, elle permet d'identifier et 'authentifier l'utilisateur. La *Private user Identity* est l'équivalent à l'IMSI pour IMS.
- MSISDN : contient une ou plusieurs numéros de téléphone pour l'utilisateur. La *Public User Identity* est l'équivalent au MSISDN pour IMS.
- CK (*Ciphering Key*) et IK (*Integrity Key*) : les clefs de chiffrement et d'intégrité utilisées pour la sécurité de l'information sur l'interface radio.
- *Long term secret* : secret utilisé pour authentifier l'utilisateur et pour générer les clefs de chiffrement et d'intégrité utilisées entre le terminal et le réseau.

- SMS : Dans ce champ sont stockés les messages courts (reçu et envoyé avec leur état).
- *SMS parameters* : paramètres de configuration du service SMS (exemple : adresse du SMS *center*).
- *MMS user connectivity parameters* : contient les paramètres de configuration du service MMS (exemple : adresse du *MMS server* et du *MMS gateway*).
- *MMS user preferences* : contient les préférences de l'utilisateur sur le service MMS comme le drapeau de rapport d'expédition.

1.4.5.2. ISIM (IMS Subscriber Identity Module)

ISIM contient les paramètres utilisés pour l'identification et l'authentification de l'utilisateur ainsi que la configuration du terminal IMS. ISIM peut coexister simultanément avec une USIM ou une SIM. Les paramètres essentiels contenus dans une ISIM sont :

- *Private User Identity*
- *Public User Identity*
- *Home Network Domain URI* : SIP URI du réseau nominal de l'utilisateur qui est unique dans la carte.
- *Long-term secret* : secret utilisé pour authentifier l'utilisateur et pour générer les clefs de chiffrement et d'intégrité utilisées entre le terminal et le réseau. Les messages SIP envoyés entre le terminal et le P-CSCF sont chiffrés et protégés par ces clefs de chiffrement et d'intégrité.

1.4.6. Signalisation en IMS

Comme SIP ne décrit pas le flux média, on utilise en plus le protocole SDP (*Session Description Protocol*) [HaPJ06]. SDP est transporté dans le cœur des messages SIP ; il décrit les sessions multimédia en termes de codeur audio, vidéo, informations de session (bande requise, type de flux…) et adressage multicast … Ces informations seront exploitées pour faire la réservation de ressource dans le plan transport.

Notons toutefois que certaines interfaces internes du réseau IMS utilisent la signalisation "*Diameter*" et non pas SIP. C'est une application standardisé par le 3GPP qui permet d'interfacer différentes entité du réseau IMS. Les échanges Diameter sont toujours du type : un message requête et une réponse associé. Les informations échangées dans ces messages sont mis dans des attributs appelés AVP (*Attribute Value Pairs*). Chaque interface Diameter a ces AVPs et ces commandes. La section suivante décrit les protocoles SIP et Diameter.

1.5. Principaux protocoles

Le cœur du domaine IMS se base essentiellement sur deux protocoles qui sont SIP et Diameter.

1.5.1. Le protocole SIP

SIP (*Session Initiation Protocol*) [RSCJ02] est le protocole fédérateur de l'architecture IMS. Il est en quelque sorte la glue qui permet aux différents composants de communiquer entre eux de manière homogène. Son choix, par rapport à tout autre protocole de signalisation, n'est pas anodin puisqu'il pérennise SIP au détriment de H.323, jugé trop lourd et trop coûteux. Il marque ainsi la confiance des industriels et du monde de la recherche dans le protocole SIP.

Bien sûr, son utilisation implique celle des protocoles associés, en particulier SDP [HaPJ06], pour la description des sessions, et RTP/RTCP [SCFJ03], pour le transport (et le contrôle) en temps réel des flux de données multimédias.

En 2002, L'IETF a rédigé une nouvelle version de SIP accompagnée d'une cinquantaine d'extensions (complexes, mais innovantes) qui ont fortement intéressé l'industrie. SIP a été retenu par les entités Packet Cable, MSF, TISPAN et l'UIT.

TISPAN a par ailleurs la mission de combler le vide créé par l'IETF dans la définition des usages des extensions de protocole SIP pour tous les cas de profils d'abonnés et d'applications à envisager dans le cadre de l'IMS [JePW02].

Au niveau d'IMS, SIP est utilisé par les équipements du réseau IMS (CSCF en particulier) pour le contrôle de l'appel (établissement, modification, fin) lors de toute session multimédia. Une session est établie dès lors que deux ou plusieurs participants échangent des données. SIP sait gérer des appels multidestinataires, par groupe ou automatiques.

1.5.1.1. Responsabilités du SIP :

SIP a la responsabilité de cinq actions :

- Déterminer la localisation de l'utilisateur ; il s'agit de son réseau mère (home) et non pas de sa localisation géographique. Avant tout établissement de session, la localisation des terminaux de bout en bout doit être faite.
- Définir les possibilités de l'appareil de l'utilisateur. L'appareil devant supporter par exemple les codecs multimédia appropriés pour pouvoir espérer une réussite dans

l'établissement de la session. En conséquence, SIP doit trouver les types de média utilisés ainsi que leurs paramètres.

- Définir les capacités de l'utilisateur. Ce à quoi l'utilisateur a droit (média, temps de session,…).
- Établir la connexion et définir les paramètres d'appel de l'appelant et de l'appelé.
- Gérer la session. Cela implique la modification des paramètres des sessions en cours, mettre fin à une session, activer les applications, …

1.5.1.2. Type de signalisation SIP :

Dans tout type de réseau, il y a toujours quatre types de signalisations :

- *Signalisation d'enregistrement* : c'est la signalisation par laquelle un terminal s'enregistre dans le réseau. Elle contient les procédures de téléchargement du profile et la gestion de la localisation. Cette signalisation est effectuée par la procédure d'enregistrement SIP (SIP REGISTER).
- *Signalisation d'appel* : c'est la signalisation par laquelle on établit une association de bout en bout entre les points d'extrémité désirant communiquer. Ce type de signalisation est caractérisé par l'échange de référence. Ceci est réalisé en IMS grâce à la procédure d'établissement de session (SIP INVITE).
- *Signalisation de connexion* : c'est l'affectation d'un service support à un appel. De proche en proche on va réserver des ressources dans le réseau selon la QoS requise pour le service. Dans SIP, cette signalisation est effectuée grâce aux entêtes SDP qui permettent de décrire le trafic et le ressources requis. Au niveau transport, on utilise les mécanismes RSVP (Ressource Reservation Protocol), DiffServ (Differentiated *Services*), pour assurer de la qualité de service dans le réseau IP.
- *Signalisation d'intelligence* : c'est la signalisation qui nous permet de faire un traitement substitutif par rapport au traitement d'appel normal. D'une façon similaire aux réseaux intelligent de type RI (INAP) ou CAMEL, les services sont exécuter par l'équivalant aux plateformes de service qui sont des serveurs d'applications (AS).

1.5.1.3. Format des messages SIP

Comme le montre la figure 1.8, un message SIP est constitué d'une ligne de démarrage, d'un certain nombre de champs d'en-tête, d'une ligne vide et d'un corps de

message qui peut être optionnel (c'est dans cette partie que l'on trouve notamment la description SDP [HaPJ06] d'une session en cours de négociation).

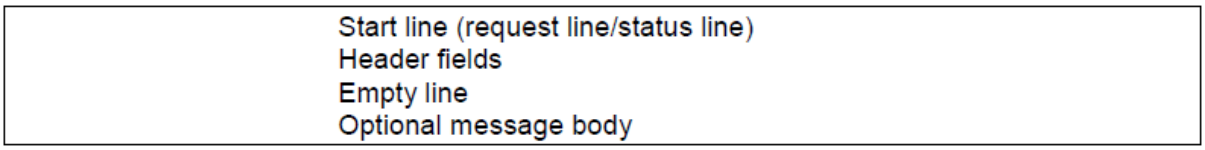


Figure 1.8. Format d'un message SIP

Les messages SIP se divisent en deux types différentiés par la *start line*: les demandes (*request line*) et les réponses (*status line*).

Les messages SIP se divisent en deux types différentiés par la ligne de démarrage (*start line)* : les demandes (*request line*) et les réponses (*status line*).

Une requête (*request line*) présente le nom de la méthode, suivi de l'URI de la destination et de la version du protocole [RSCJ02]. Il existe plusieurs méthodes SIP que l'on retrouve dans le tableau 1.1 (*Certaines méthodes présentées dans ce tableau ont été introduites dans le cadre d'extensions de SIP, comme PRACK*).

Une ligne de statut (*status line*) est constituée d'un numéro de version et du statut de la transaction sous la forme d'un code et d'une phrase. Les codes des différents statuts sont compris entre 100 et 699, classés par centaine (tableau 1.2).

Mis à part cette première ligne, requêtes et réponses sont constituées de la même manière. Dans l'exemple suivant, nous illustrons une requête SIP qui est envoyée à Alice par Bob. Bob utilise la méthode INVITE pour demander un nouvel appel. L'entête *via* piste le chemin de l'appelant vers l'appelé. Ici il s'agit de l'UA (pour *User Agent*) de Bob. Le champ *call-ID* est un identifiant unique pour suivre cet appel, il a été généré par l'UA appelant. *CSeq*est utilisé pour compter les messages dans un appel et ainsi appairer requêtes et réponses (figure 1.9).

Tableau 1.1. Liste des différentes méthodes SIP

Méthode	Utilisation
ACK	Acquittement de la réponse définitive à un message INVITE
BYE	Fin d'une session
CANCEL	Annulation d'une requête en attente
INFO	Message contenant de la signalisation téléphonique (PSTN)
INVITE	Demande d'ouverture de session
NOTIFY	Notification d'événement à un client SIP
OPTIONS	Demande d'information sur les capacités d'un serveur
PRACK	Acquittement d'une réponse provisoire à un message INVITE
PUBLISH	Exportation d'informations vers un serveur
REGISTER	Enregistrement de la correspondance entre AoR et URI courante

SUBSCRIBE	Demande d'enregistrement à un service de notification
UPDATE	Modification des caractéristiques d'une session en cours
MESSAGE	Message instantané
PEFER	Demande à un serveur d'envoi d'une requête

Tableau 1.2. Liste des différents types de codes retours

Codes	Types
De 100 à 199	Information
De 200 à 299	Succès
De 300 à 399	Redirection
De 400 à 499	Erreur – niveau client
De 500 à 599	Erreur – niveau serveur
De 600 à 699	Erreur globale

```
INVITE sip:bob@enseeiht.fr SIP/2.0
Via: SIP/2.0/UDP yon.enseeiht.fr;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Alice <sip:alice@irit.fr>
From: Bob <sip:bob@enseeiht.fr>;tag=1928301774
Call-ID: a84b4c76e66710@yon.enseeiht.fr
CSeq: 314159 INVITE
Contact: <sip:bob@enseeiht.fr>
Content-Type: application/sdp
Content-Length: 142
(Bob's SDP not shown)
```

Figure 1.9. Exemple de requête SIP d'invitation

1.5.2. Le protocole Diameter

Diameter [AZFL12, CLGZ03] est un protocole permettant à des domaines administratifs différents de collaborer pour réaliser les fonctionnalités AAA (*Authentication, Authorization, Accounting*). Il est constitué d'un protocole de base qui définit le format des messages, comment ils sont transportés, les messages d'erreurs, ainsi que les services de sécurité que toutes les implémentations doivent supporter.

Diameter a été conçu dans l'idée d'être facilement extensible. C'est une version améliorée de Radius qui offre notamment l'avantage de mobilité. Dans un domaine IMS, on utilise Diameter entre le HSS et l'I/S_CSCF pour la gestion des autorisations des abonnés.

1.5.2.1. Les responsabilités de Diameter

Tel qu'il a été défini par l'IETF, Diameter peut être divisé en deux composants :

- Le protocole Diameter en lui-même (Diameter Base) : fournit les conditions de communication et de format des messages, et utilise les protocoles TCP ou SCTP. Ses fonctions sont essentiellement :
    - Livraison des AVPs (*Attibute-Value Pair*) ;

- Négociation des capacités ;
- Notification des erreurs ;
- Gestion de session.

- Diameter Application : placée au-dessus du "Diameter Base", cette partie du protocole hérite des facilités de communication ainsi offertes, fournit des messages spécifiques de requête/réponse pour les services utilisés. Ses fonctions sont :
  - Autorisation de l'utilisateur ;
  - Assignation d'un serveur (S-CSCF) ;
  - Récupération des informations de localisation ;
  - Authentification multimédia ;
  - Finalisation de l'enregistrement ;
  - Modifications du profil de l'utilisateur.

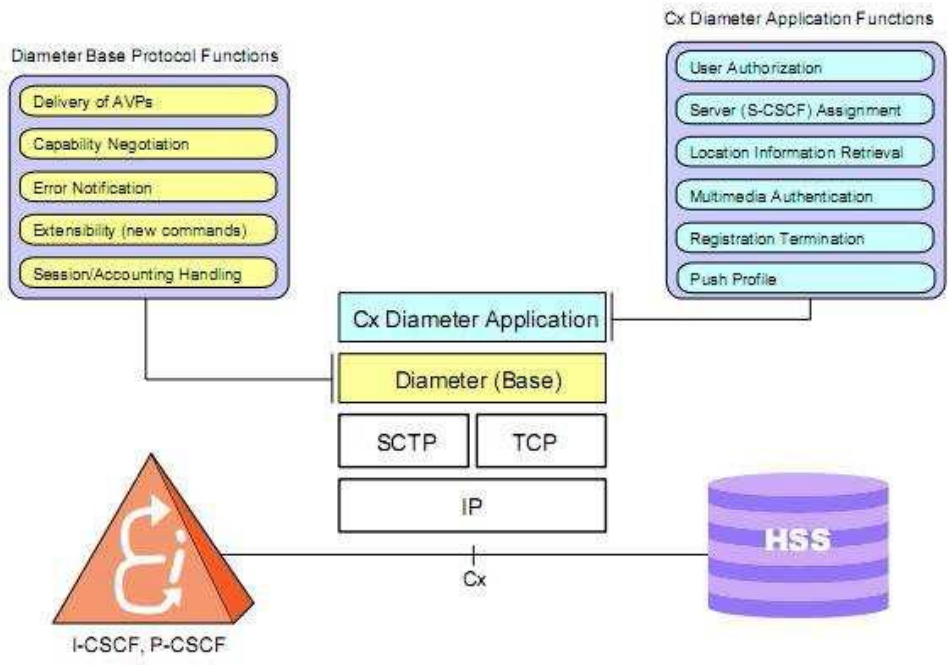


Figure 1.10. Niveaux du protocole Diameter (base et application)

1.5.2.2. Sa structure de message

Tout message Diameter utilise le schéma montré dans la figure 1.11 :

1. *Version* : version du protocole utilisée.
2. *Message Length* : taille du message, incluant l'entête.
3. *Command Flags* : trois bit, le premier indique la nature du message (requête ou réponse), le second indique si le message doit être transféré ou traité localement, le dernier est

utilisé seulement pour les réponses importantes (peut spécifier si la réponse contient une erreur).

4. *Command Code* : contient un indicateur d'appartenance du message.
5. *Application ID* : identifie l'application Diameter utilisée.
6. *Hop-by-Hop ID* : permet d'associer la réponse à la requête.
7. *End-to-End ID* : identifiant du message qui doit rester le même tout le long, afin d'éviter les duplications. Cet identifiant doit donc rester unique.
8. *AVP* : contient les paires attribut-valeur (ou tuple) constituant les paramètres du message.

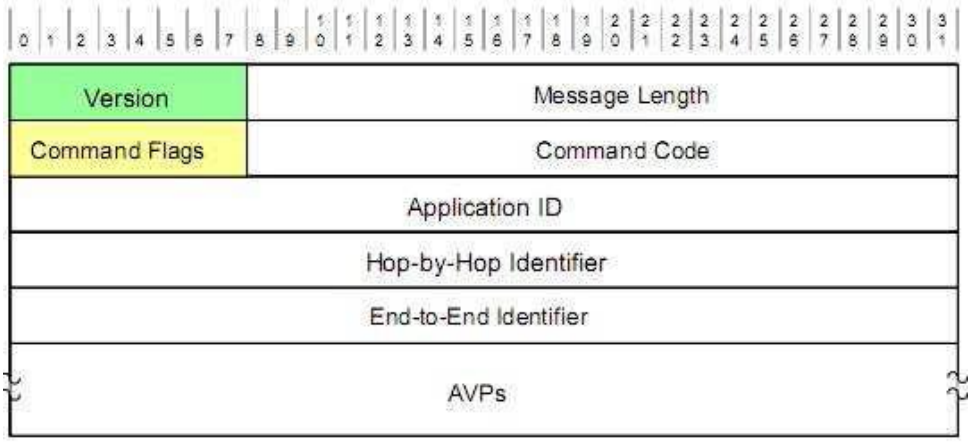


Figure 1.11. Structure du message Diameter

1.6. Conclusion

L'objectif de ce chapitre est de fournir un point de départ pour comprendre les normes entourant IMS, un sujet immense et très compliqué. Nous avons présenté les concepts de l'architecture IMS, sa motivation ainsi que les bases des différents protocoles utilisés (SIP, Diameter). Pour bien identifier les problèmes et les vulnérabilités de sécurité dans une architecture IMS, et ainsi proposer des contre-mesures et des améliorations à ces problèmes, nous allons présenter dans le chapitre suivant les différentes attaques visant IMS. Nous présentons aussi les différents mécanismes de sécurité proposés par la norme.

Chapitre 2 : Sécurité en IMS

Sans doute, la convergence des réseaux voix et données est une grande réussite pour maintenir une plateforme de communication unique pour tous. Cela dit, le plus grand défi est de maintenir un niveau adéquat de sécurité assurant l'intégrité et la confidentialité des données ainsi que la disponibilité des services.

L'IMS est en particulier vulnérable aux différents attaques « *peer-to-peer* » vu l'utilisation du protocole SIP pour la signalisation qui est une architecture ouverte.

Les menaces dans l'IMS comprennent également des attaques par inondations, qui rendent les ressources réseau occupées. Les serveurs d'applications IMS, qui fournissent des services à valeur ajoutée, sont également des cibles précieuses pour les intrus. En raison de la nature « *textbased »* de SIP, les ASs sont vulnérables aux attaques de type usurpation (*spoofing*) et falsification de message. Enfin, les intrus peuvent lancer un déni de service (DoS) contre des applications installées sur l'AS.

Dans ce second chapitre, nous présenterons les défis de sécurité en IMS et les différentes attaques potentielles. Ensuite nous décrivons l'architecture de sécurité en IMS et les associations de sécurité, ainsi que les différents mécanismes de sécurité. Puis nous terminons ce chapitre par une conclusion.

2.1. Défis de sécurité en IMS et attaques potentielles

Les problèmes de sécurité face à IMS [ShMa07, WaLi09] sont des menaces venant de différents protocoles par exemple, les attaques visant la signalisation SIP, les attaques RTP des médias, et les attaques du domaine IP (Figure 2.1).

Pour identifier les principales vulnérabilités et les menaces de l'architecture IMS, nous avons utilisé une méthode rigoureuse et systématique [MaKO11] basée sur la recommandation l'UIT-T X.805. La Recommandation UIT-T X.805 [Inte03] peut être utilisé pour augmenter les spécifications de sécurité et donc fournir une sécurité globale de bout-en-bout au niveau du réseau IMS en incluant le plan de contrôle, le plan de gestion, ainsi que les plans de

l'utilisateur final. La Recommandation UIT-T X.805 définit huit mesures de sécurité pour se protéger contre les différentes menaces :

- Contrôle d'accès;
- Authentification;
- Non-répudiation;
- Confidentialité des données;
- Sécurité de la communication;
- Intégrité des données;
- Disponibilité;
- Respect de la vie privée.

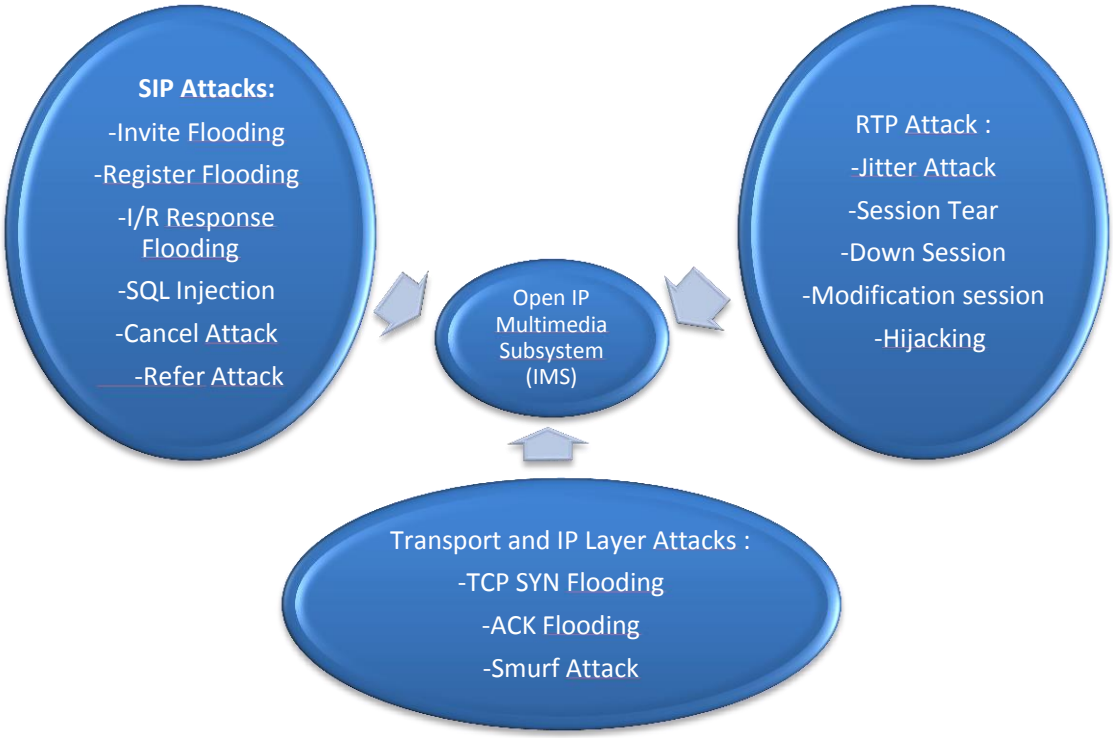


Figure 2.1. Les menaces potentielles en IMS

Cependant, les huit dimensions définies dans la Recommandation UIT-T X.805 ne sont pas appropriés pour l'analyse de l'IMS. Prenons par exemple les deux dimensions de contrôle d'accès et d'authentification, le contrôle d'accès dépend de l'authentification pour assurer son objectif en matière de sécurité, et des techniques d'authentification peuvent être nécessaires dans le cadre du contrôle d'accès. Nous remarquons qu'il y a une intersection entre ces deux dimensions. Pour étudier la sécurité d'IMS d'une manière claire et systématique, le modèle

adopté doit s'assurer que la dimension qui porte sur un aspect particulier de la sécurité est orthogonale et complète. Le même raisonnement peut être fait pour la confidentialité et la protection de la vie privée. Par conséquent, six dimensions de sécurité sont identifiées. Le tableau 2.1 fournit ces dimensions de sécurité avec les principales menaces, relatives au réseau IMS, qui les atteignent.

Tableau 2.1. Dimensions de sécurité et menaces correspondantes

Dimension de Sécurité	Menace de Sécurité
Authentification	Usurpation
Non-Repudiation	Vol (suppression)
Confidentialité	divulgation
Sécurité de la communication	Accès non autorisé
Integrité	Falsification des données
Disponibilité	Destruction / interruption

Globalement, nous classons les menaces de sécurité IMS en trois catégories. Tout d'abord, les attaques communes aux applications sur le réseau Internet telles que les attaques par rejeu, le déni de service, l'usurpation (*spoofing*), le reniflage (*sniffing*) et autres attaques par interception. Il y a ensuite des attaques spécifiques à la nature du protocole SIP (à la couche application). De par la structure de ses messages, SIP est vulnérable à certaines attaques telles que de fausses inscriptions ou terminaisons de sessions. Les vulnérabilités de SIP ont été relevées dès sa conception dans le RFC 3261 (celui-ci indique de façon générale les problèmes) mais sans qu'il y ait de solutions proposées. Finalement nous pouvons considérer, qu'en raison de la jeunesse et la complexité des applications reposant sur SIP (VoIP et le multimédia), les serveurs et autres produits SIP sont susceptibles de souffrir de vulnérabilités connues mais non forcément corrigées telles que l'injection SQL et le dépassement de tampon mémoire. Les principales menaces sont indiquées ci-après :

- Déni de service (DoS) : attaque contre la disponibilité

Des signaux radio et des inondations par les demandes d'authentification au P-CSCF et d'autres dispositifs. Par exemple, lors d'une attaque d'inondations REGISTER, l'attaquant envoie de nombreuses demandes REGISTER au P-CSCF avec des adresses sources fausses ou falsifiées (SIP URI). Dans le cas des inondations REGISTER distribués, l'attaquant génère

de multiples demandes REGISTER avec différentes adresses source usurpées et truquées pour submerger les ressources IMS. Il provoque ainsi la chute des ressources IMS et les utilisateurs légitimes ne peuvent pas obtenir des services.

- Usurpation (Spoofing) : attaque contre l'authentification

Le nœud malveillant se présente dans le réseau et intercepte le trafic, ce qui permet aux attaquants d'altérer les messages.

- Phishing : attaque contre l'authentification, la confidentialité et l'intégrité

Les attaquants peuvent usurper les identifiants utilisateur dans leurs connexions avec le serveur SIP pour voler des informations ou effectuer des actions malveillantes. Par exemple, l'attaquant peut enregistrer son proxy SIP, avec un nom semblable à un organisme de la victime, sur le site web proxy SIP de localisation. Lors du lancement d'une session SIP, la victime sera en réalité en contact avec l'attaquant du serveur SIP (au lieu de son serveur SIP légitime). L'attaquant sera alors en mesure d'obtenir des informations sur les victimes. Par ailleurs, l'attaquant peut usurper l'identité d'une entité SIP, par exemple l'identité de l'organisation de confiance et appeler la victime pour lui demander des informations afin de commettre acte un malveillant.

- Fin de session: attaque contre la disponibilité du service en cours

Un attaquant pourrait utiliser la requête BYE pour mettre fin à une session. Le pirate prétend utiliser UE2 et envoie un faux message BYE du P-CSCF au UE1. UE1 arrête ainsi d'envoyer le flux RTP immédiatement, tandis qu'UE2 continue à envoyer des paquets RTP à UE1. Pour lancer ce genre d'attaque, l'attaquant a besoin de connaitre tous les paramètres de session. Ces informations peuvent être obtenues en sniffant le réseau ou en effectuant une attaque man-in-the-middle pour insérer une requête BYE dans la session comme illustré en figure 2.2.

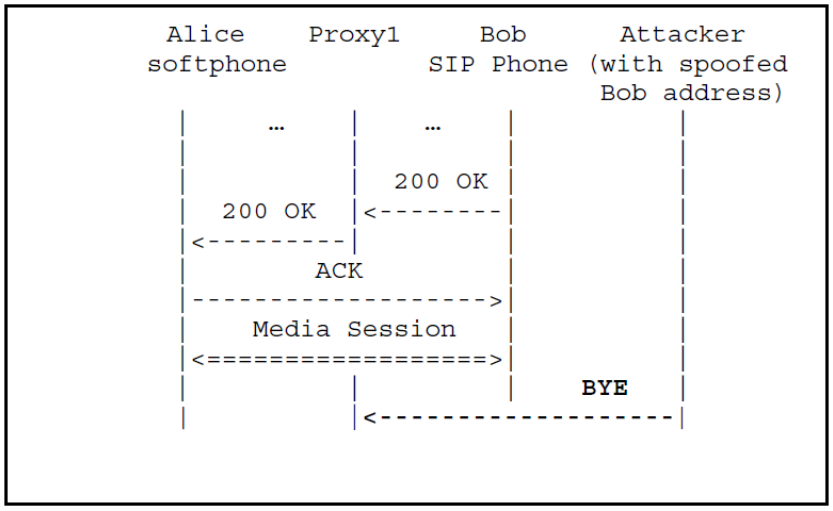


Figure 2.2. L'attaquant envoie un message BYE pour mettre fin à la connexion

- Attaque CANCEL: attaque contre la disponibilité du service en cours

Cette attaque met fin à une demande en attente. L'attaquant pourrait utiliser la méthode CANCEL pour annuler une demande INVITE générée par un utilisateur légitime. Avant que la réponse finale soit générée pour une demande INVITE, l'attaquant envoie un « faux » message CANCEL au P-CSCF tout en laissant croire qu'il provient d'un utilisateur légitime. Le noyau IMS accuse la réception du message CANCEL et cesse le traitement de la demande INVITE.

- Attaque Re-INVITE: attaque contre la disponibilité du service en cours

La requête INVITE établit une session ou un dialogue entre deux utilisateurs (UE). L'objectif du message re-INVITE est de modifier les informations de la session actuelle, par exemple, en changeant les adresses ou les ports, l'ajout d'un flux de support, ou la suppression d'un flux de support. Par conséquent, l'attaquant pourrait lancer une attaque DoS en envoyant un message re-INVITE forgé pour modifier la session.

- Deviner le mot de passe : attaque contre l'authentification

Ressemble à l'attaque de détournement de session avec l'objectif d'obtenir des informations sur la session de l'utilisateur. Même si un intrus n'est pas capable de briser le processus d'authentification IMS, cette attaque pourrait être lancée pour une mauvaise utilisation des comptes d'utilisateurs légitimes. L'intrus lance cette attaque en envoyant de nombreuses demandes REGISTER au P-CSCF et reçoit des messages « 401- unauthorized » de réseau IMS. Si l'attaque réussie, l'attaquant obtient une réponse « 200 OK ».

- Voice trapping : attaque contre la confidentialité

Comme le trafic de voix sur IP est envoyé sur le réseau IP sans aucun chiffrement, l'espionnage des paquets est ainsi possible, en particulier dans les réseaux sans fil, où il est plus facile de piéger les données (par rapport au réseau téléphonique traditionnel qui est point à point, non diffusé).

- Message mal formé : attaque contre la disponibilité, la confidentialité et l'intégrité

Pour exécuter des codes malveillants ou pour afficher des données confidentielles (par exemple, les utilisateurs de login / mot de passe), un attaquant peut provoquer un dépassement mémoire (Figure 2.3) ou une attaque par injection de code SQL (Figure 2.4).

```
INVITE sip:bob@enseeiht.fr SIP/2.0
Via: SIP/2.0/UDP 147.127.240.x:5060
From: %s%s%s%s%s%s%s%s%s%s%s%s%s%s%s <sip:alice@enseeiht.fr>; tag=765809
To: Bob <sip:bob@enseeiht.fr>
Call-ID: 8604553107@enseeiht.fr
CSeq: 1 INVITE
Contact: <sip:alice@147.127.240.x>
Content-Type: application/sdp
Content-Length: 138
:
```

Figure 2.3. Exemple d'un dépassement de tampon dans le cadre du protocole SIP

```
Original register message:
REGISTER sip:sipdemo.enseeiht.fr SIP/2.0
From: Alice <sip:alice@sipdemo.enseeiht.fr>
:
SQL lookup: select * from users where username="alice";

SQL injection message:
REGISTER sip:sipdemo.enseeiht.fr SIP/2.0
From: Alice <sip:alice";drop table users;--@some.sip.domain>
:
SQL lookup: select * from users where username="alice"; drop table users;--;
```

Figure 2.4. Exemple d'une injection SQL dans le cadre du protocole SIP

Par ailleurs, l'interface web sur le serveur téléphonique central pour le téléphone SIP peut être vulnérable à une attaque par script croisé (cross-site scripting: XSS) qui résulte en une action ne correspondant pas à celle qui devrait être entreprise par la page web. Une attaque XSS permet à des attaquants distants d'injecter un script web malicieux, ou des codes HTML malicieux via le champ "From" dans un message SIP.

- Vol de medias : attaque contre non-répudiation

Dans un environnement de raccordement fixe, une fois la session est établie entre deux terminaux, les flux de médias peuvent être échangés directement entre les deux terminaux (sans passer par les CSCFs). Dans un fonctionnement normal, lorsqu'un terminal envoie ou reçoit une requête *SIP BYE*, il va libérer les médias actifs. Mais si on modifie les deux terminaux de tel sorte que quand ils reçoivent une requête *SIP BYE* ils ne libèrent pas les canaux médias. De cette façon, lorsqu'un des deux terminaux envoie la requête *SIP BYE* aux CSCF, CSCF pense que la session terminée et va arrêter la comptabilité. Les deux terminaux peuvent donc continuer la communication, et ainsi voler des ressources médias. Un exemple est illustré en Figure 2.5.

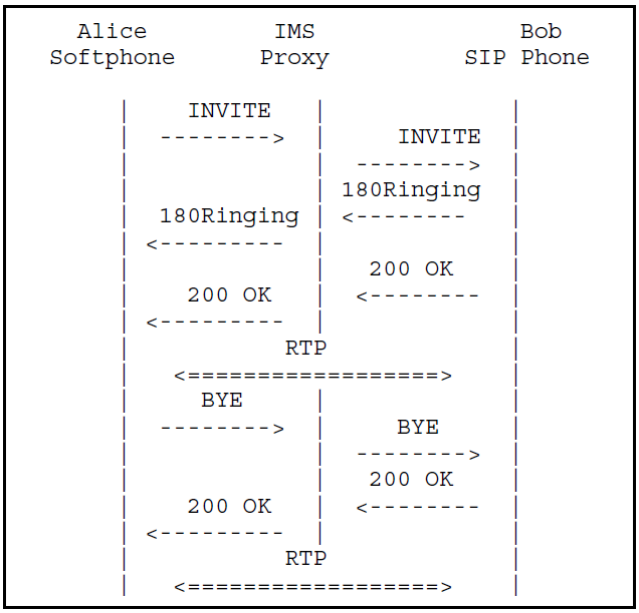


Figure 2.5. Exemple d'un scénario de vol de médias

Après avoir présenté une panoplie d'attaques pouvant être utilisées dans un environnement IMS, nous allons détailler dans la section suivante comment s'en protéger.

2.2. Mécanisme de sécurité en IMS

La sécurité IMS est divisée en sécurité des accès (spécifié dans [Gpp09a]) et sécurité du réseau (spécifié dans [Gpp09b]).

- La sécurité des accès comprend l'authentification mutuelle des utilisateurs et du réseau, ainsi que la protection du trafic entre le terminal et le réseau IMS (P-CSCF).

- La sécurité du réseau traite de la protection du trafic entre les nœuds du réseau, qui peuvent d'ailleurs appartenir au même opérateur ou à différents opérateurs.

Détaillons dans les sous-sections suivantes les principaux mécanismes (essentiellement pour l'authentification, la confidentialité et l'intégrité) utilisés pour sécuriser les accès et les réseaux.

2.2.1. Sécurité des accès

2.2.1.1. Authentification

Le mécanisme de base (expliqué en détail dans le chapitre 4) utilisé pour l'authentification dans IMS est IMS-AKA (*Authentication and Key Agreement*) [Gpp09c, Gpp09d]. Celui-ci repose sur l'utilisation d'un secret partagé entre l'utilisateur (sur l'ISIM ou à défaut l'USIM) et le réseau IMS (au niveau du HSS). Le HSS, toujours localisé dans le réseau "home" de l'utilisateur, contient un quintuple d'authentification :

1. un challenge aléatoire, RAND ;
2. un jeton d'authentification, AUTN (contenant un numéro de séquence et un identifiant MAC) ;
3. une réponse attendue XRES ;
4. une clé de chiffrement, CK, et une clé d'intégrité, IK, qui assurent la confidentialité et l'intégrité des messages SIP échangés entre le terminal et le P-CSCF.

Durant l'enregistrement, le S-CSCF transmet les paramètres RAND et AUTN au terminal qui s'en sert pour calculer une réponse attendue XMAC. Précisons que l'authentification est mutuelle entre l'utilisateur et le réseau (S-CSCF) :

- le terminal compare son XMAC avec le MAC reçu du réseau : le réseau est authentifié si la valeur reçu et celle calculée sont identiques ;
- le réseau compare son XRES avec le RES calculé par le terminal : l'utilisateur est authentifié s'ils sont identiques.

Il est à noter que l'authentification dans l'IMS réutilise les principes d'authentification de l'UMTS (UMTS AKA). Aussi, certains mécanismes de l'UMTS-AKA peuvent être partagés avec IMS-AKA. En particulier, tous les mécanismes de l'UMTS-AKA sont réutilisés si le terminal contient une USIM (équivalent de la carte SIM pour l'UMTS) et non une ISIM (équivalent de la carte SIM pour l'IMS), à savoir :

- la clé d'authentification K qui sert à calculer les paramètres d'authentification (XMAC, RES, IK, CK) ;

- les algorithmes de sécurité pour calculer les clés de sécurité ;
- le mécanisme de vérification du numéro de séquence qui sert à synchroniser les clés de sécurité entre le terminal et le réseau.

Lorsque seule l'USIM est utilisée, il faut dériver les paramètres utiles à l'IMS (qui seraient normalement sur l'ISIM) depuis l'IMSI (identifiant de l'utilisateur pour l'UMTS) stocké sur l'USIM :

- l'identité privée de l'utilisateur qui est une identité publique temporaire qui ne sert que lors de l'enregistrement, les identités publiques associées peuvent en revanche être utilisées ;
- une identité publique de l'utilisateur ;
- le nom de domaine.

2.2.1.2. Tunnel IPSec entre terminal et P-CSCF

Les messages de signalisation SIP sont protégés grâce à la montée d'associations de sécurité [KeSe05a] entre le terminal et le P-CSCF. Les clés utilisées pour les associations de sécurité sont CK et IK :

- l'intégrité des données (qui garantit à la fois l'authentification de l'émetteur ainsi que la non altération des messages SIP reçus) est obligatoire en Rel-5/Rel-6. Cette propriété est assurée grâce à l'utilisation de tunnel IPSec [Kent05] ;
- la confidentialité des données est obligatoire en Rel-6 seulement.

2.2.2. Sécurité dans le réseau de cœur : Architecture NDS

2.2.2.1. Passerelles de sécurité

Des passerelles de sécurité (SEG pour *Security Gateway*) sont introduites à la frontière entre deux domaines de sécurité NDS (Network Domain Security) [Gpp09b]. Une protection de type IPSec ESP en mode tunnel est obligatoire entre deux passerelles de sécurité (IPSec peut assurer l'intégrité et la confidentialité et l'anti-rejeu) tandis qu'elle est optionnelle entre deux nœuds d'un même domaine de sécurité.

Typiquement, lorsque le P-CSCF, I-CSCF et S-CSCF appartiennent au même réseau, la protection IPSec de leurs échanges est optionnelle. En revanche, un échange entre deux CSCF d'opérateurs différents doit être protégé par IPSec (ESP en mode tunnel) comme l'illustre la figure 2.6.

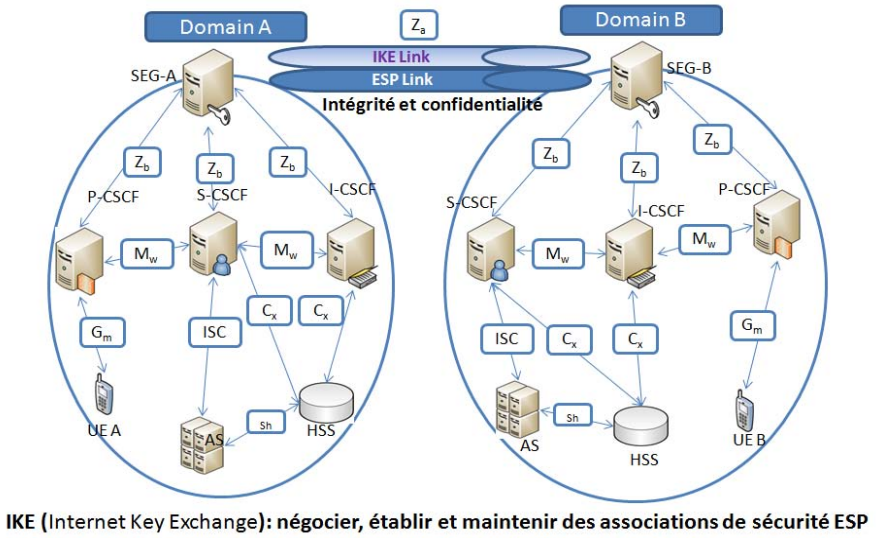


Figure 2.6. Architecture de passerelle de sécurité

Ces deux modes de protection utilisés pour assurer la sécurité du réseau de cœur ne sont que les deux interfaces Za et Zb définies par 3GPP. Zb est utilisée à l'intérieur du même réseau, elle est facultatif et dépend de l'administrateur du domaine de sécurité. Par contre, l'interface Za est obligatoire et est utilisée entre des domaines de sécurité différents. L'authentification et l'intégrité des données sont obligatoires pour les deux interfaces.

Il est à noter également qu'un échange entre une entité de l'IMS et une entité d'un réseau non IMS peut être protégé par TLS (Transport Layer Security) [Stur11], typiquement pour l'interfonctionnement en Rel-6 de SIP entre le S-CSCF dans l'IMS et un proxy SIP externe dans un réseau non IMS (tunnel TLS sur la figure 2.7). Le S-CSCF gère localement une liste de partenaires pour l'interfonctionnement, ce qui lui permet de décider s'il peut faire confiance à ce proxy SIP.

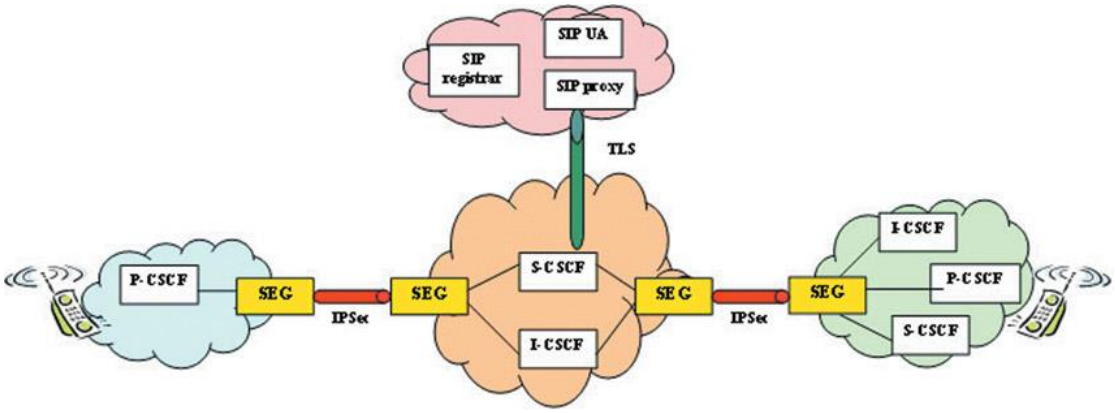


Figure 2.7. Sécurité dans le réseau de cœur IMS

2.2.2.2. Fonction I-CSCF (THIG)

La fonction THIG est implémentée en option à l'intérieur du I-CSCF et a pour rôle de chiffrer/déchiffrer les en-têtes SIP qui renseigneraient sur la topologie du réseau "home", par exemple les en-têtes "Via", "Route", "Record-Route", "Path" qui montreraient les adresses des nœuds de réseau traversés par une requête SIP.

2.2.3. Framework de sécurité proposé par le 3GPP2

2.2.3.1. Objectifs des solutions de sécurité IMS

Dans la vision 3GPP2 l'objectif des solutions de sécurité IMS est de développer un *framework* de sécurité IMS pour assurer la confidentialité de l'utilisateur et la protection du réseau contre les utilisations abusives. Les importants caractéristiques et services de sécurité sont fournis par ces solutions:

La confidentialité de l'utilisateur fournit la confidentialité de l'identité, la confidentialité de l'emplacement et la non-traçabilité de l'utilisateur. Pour parvenir à ces caractéristiques, l'utilisateur se voit attribuer une identité temporaire de sorte que l'identité permanente de l'utilisateur, dans laquelle les services sont fournis ne peut pas être espionné via le lien d'accès radio. En outre, les données de l'utilisateur et la signalisation qui pourraient révéler l'identité de l'utilisateur sont chiffrées sur le lien d'accès radio.

L'authentification de l'entité est basée à la fois sur l'authentification des utilisateurs et sur l'authentification réseau et devrait s'appliquer à la configuration de la connexion entre l'utilisateur et le réseau. Il s'agit d'un mécanisme d'authentification en utilisant un vecteur d'authentification délivré par le réseau de l'utilisateur (*Home Network*) au le réseau de service (réseau par lequel l'utilisateur est connecté), mais aussi en utilisant un mécanisme d'authentification local en utilisant la mise en place des clés d'intégrité entre l'utilisateur et le réseau de service.

La confidentialité des données des utilisateurs et des données de signalisation. Elle est réalisée en utilisant des algorithmes de chiffrement et d'un accord sur les clés (*Key Agreement*).

L'intégrité des données et l'authentification de l'origine des données de signalisation. L'intégrité des données nécessite bien évidement un accord sur les algorithmes et les clés (d'intégrité) à utiliser.

Disponibilité du réseau et ses services permet de s'assurer que les ressources réseau et les services sont disponibles tout le temps pour les utilisateurs. Pour assurer la disponibilité de services et de ressources, le réseau doit être protégé contre les attaques par déni de service (DoS) et déni de service distribué (DDoS) [Chen06, ReSM08].

Contrôle de la fraude protège les ressources précieuses et les services à valeur ajoutée des utilisateurs illégitimes et des pirates. En IMS, ces services peuvent être protégés en sécurisant les associations de sécurité (ASs). Le 3GPP et 3GPP2 ont normalisé la sécurité IMS dans différentes versions et sont basés sur :

- Une solution normalisée par 3GPP version 5 appelée « early IMS » qui offre des fonctionnalités de sécurité limitée et vise à protéger le déploiement rapide d'IMS, elle offre à ce titre moins de sécurité. Les services offerts sont essentiellement l'authentification des abonnés pour l'accès aux services, la confidentialité de l'identité sur l'interface radio ainsi que le chiffrement de l'interface radio.
- Une solution complète pour la sécurité d'IMS normalisée par 3GPP version 6 avec des fonctionnalités évoluées de sécurité, elle s'appuie sur les solutions de sécurité « early IMS » avec un objectif de les améliorer. Elle offre de nouvelles fonctionnalités de sécurité et sécurise de nouveaux services pour protéger les réseaux et les terminaux. Elle se compose de la sécurité du domaine du réseau et de la sécurité d'accès qui définissent la sécurité SIP en mode *hop-by-hop*. La sécurité de bout en bout n'est donc pas prise en charge. La sécurité globale d'IMS comprend les mécanismes suivants et est représentée dans la figure 2.8 :
    - authentification et échange de clé entre un abonné IMS et son réseau mère;
    - accord sur les mécanismes de sécurité entre le client IMS et le réseau visité;
    - protection et confidentialité de l'intégrité;
    - sécurité du domaine réseau entre différents domaines, et
    - sécurité d'accès GPRS/UMTS existante.

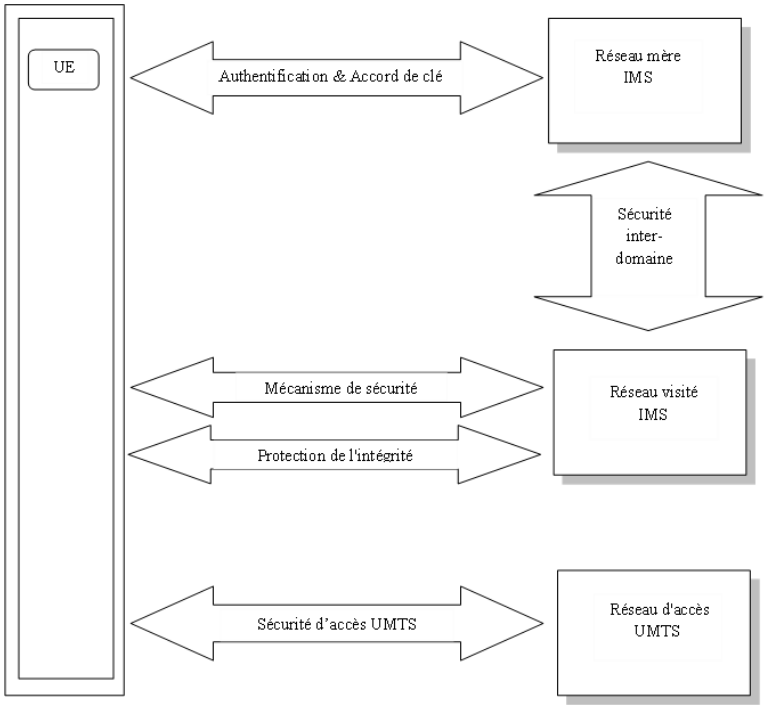


Figure 2.8. Vue globale de la sécurité en IMS

2.2.3.2. Associations de sécurité

Nous énumérons sept associations de sécurité différents et des besoins différents en matière de protection de sécurité pour IMS (y compris les Serveurs d'Applications SIP – SIP AP) ; ils sont numérotés de 1 à 7 dans la figure 2.9 [ShCM08].

AS1 assure l'authentification mutuelle entre l'UE et le S-CSCF. Le HSS délègue la gestion de l'authentification des abonnées au S-CSCF, cependant le HSS est responsable de la génération des clés et des défis. La clé à long terme pré-partagée dans l'ISIM et le HSS est associée à l'identité privée de l'utilisateur (IMPI).

AS2 fournit un lien sécurisé et une association de sécurité entre l'UE et le P-CSCF pour protéger le point de référence Gm (contact avec l'air). L'Authentification de l'origine des données est également assurée.

AS3 assure la sécurité dans le domaine du réseau interne au niveau de l'interface Cx. Le protocole utilisé au niveau de cette interface est DIAMETER, sécurisé par IPsec.

AS4 assure la sécurité entre différents réseaux. Cette association sécurise les échanges entre les nœuds SIP, et est applicable uniquement lorsque P-CSCF réside dans le réseau visité (c.-à-d. utilisateur est en itinérance).

AS5 assure la sécurité au sein du réseau interne entre les serveurs SIP et s'applique aussi lorsque le P-CSCF se trouve dans le *Home Network*. L'IMS protège tout le trafic IP dans le cœur du réseau en utilisant NDS/IP [Gpp09b], qui fournit la confidentialité, l'intégrité des données, l'authentification et la protection anti-rejeu pour le trafic. Le mécanisme de sécurité appliqué est le protocole IPSec.

SA6 assure la sécurité entre un nœud SIP résidant dans un réseau IP externe et le HSS. Par un nœud SIP on veut dire un serveur d'applications SIP (SIP AS) qui peut également être hébergé dans le même *Home Network*. Cependant l'association SA6 est applicable uniquement lorsque le SIP AS réside dans un réseau IP externe (sinon c'est l'AS3 qu'il faut appliquer).

AS7 assure la sécurité entre les nœuds SIP situés dans différents réseaux. Elle diffère de l'AS4 dans la mesure où le nœud SIP ici est un serveur d'applications SIP. Elle est applicable uniquement lorsque le SIP AS réside dans un réseau IP externe (sinon c'est l'AS5 qu'il faut appliquer).

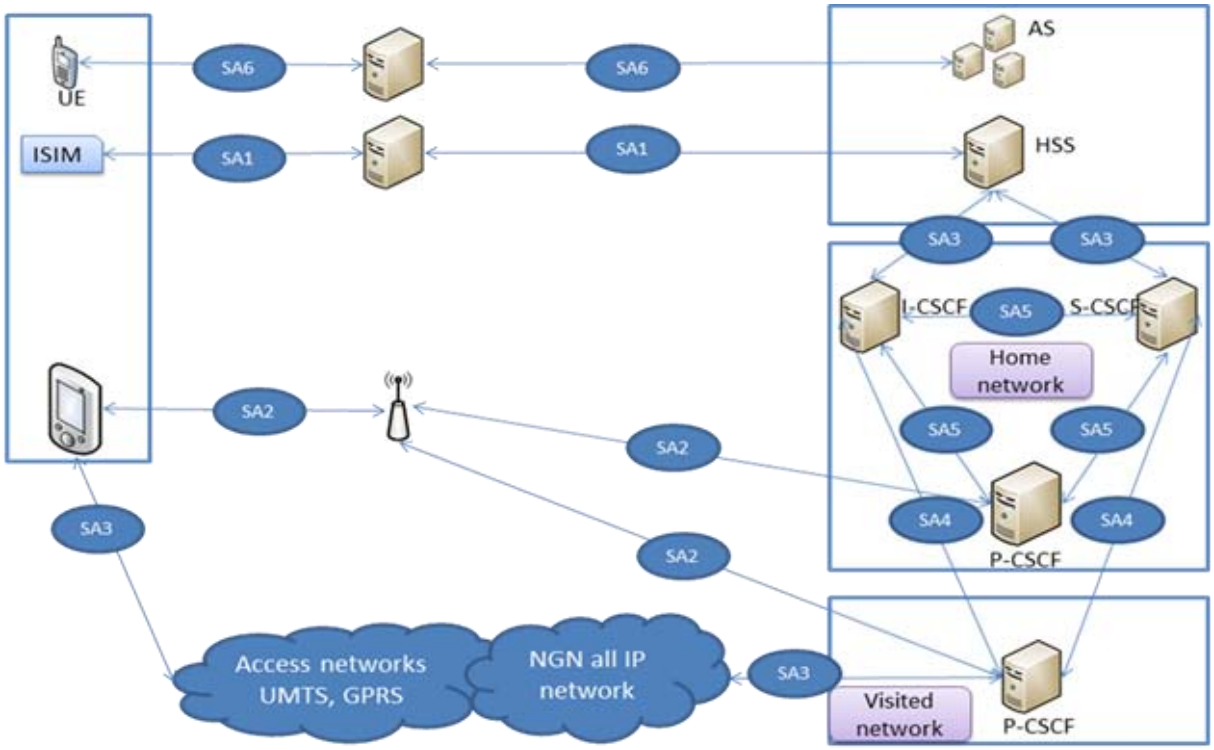


Figure 2.9. Association de sécurité IMS

2.3. Conclusion

L'IMS se pose à la frontière entre deux paradigmes : le monde des réseaux IP et celui des réseaux télécoms. En décrivant une architecture unique, l'IMS permet la cohésion de deux

réseaux qui étaient, jusqu'à présent, indépendants, voire concurrents sur de nombreux services, comme la téléphonie.

Telle est la prétention de l'IMS : quel que soit le réseau d'accès, faire émerger un réseau cœur unique, qui mutualise toutes les ressources, standardise les échanges entre les opérateurs, unifie les services des utilisateurs, simplifie la gestion des terminaux et facilite le développement de nouveaux services. C'est un pas nécessaire dont la grande majorité des acteurs reconnaissent la nécessité. C'est néanmoins une étape que peu d'acteurs osent encore franchir.

En effet, si son utilité n'est pas à démontrer, le business plan de l'IMS (et donc sa rentabilité) reste à étudier. Par exemple, la première des applications souvent mentionnées pour l'IMS est le service de présence, comme sur les messageries instantanées. Mais, avec ce simple service, les opérateurs peuvent perdre énormément d'appels, en particulier ceux qui aboutissent sur la messagerie. Or l'infrastructure d'un réseau IMS a un coût relativement important pour les opérateurs, qu'il faut rentabiliser à juste titre. Ce point sera traité au niveau du chapitre 5 par la proposition d'un nouveau modèle de provisionnement de service. Ce modèle permettra d'offrir une nouvelle forme de consommation à forte valeur ajoutée aux clients et ainsi d'augmenter la rentabilité de l'IMS.

Une autre préoccupation majeure de l'IMS est inévitablement la sécurité totale mais à moindre coût. Le chapitre 4 s'intéresse à ce besoin et propose un nouveau protocole d'authentification plus sécurisé que celui utilisé en IMS (IMS-AKA) et plus performant en termes d'utilisation de la bande passante et de temps de traitement. L'étude du protocole IMS-AKA sera divisée en deux parties, en chapitre 3 nous présentons les différents mécanismes de d'authentification ainsi que les concepts de base de la sécurité que nous utilisons dans cette étude. Ensuite, dans le chapitre 4, nous étudions les vulnérabilités du protocole IMS-AKA, nous présentons notre solution avant de nous intéresser à l'implémentation et l'évaluation de la solution.

Chapitre 3 : Mécanismes d'authentification

Que ce soit pour un accès à des réseaux locaux ou étendus, que ces réseaux soient filaires ou sans fil, que ces réseaux soient en architecture client-serveur ou répartie, l'authentification des équipements, des services et des personnes est nécessaire. Tout ce qui concerne l'accès privé, c'est-à-dire le contrôle de la délivrance de l'information et de la fourniture des ressources réservées à certaines entités, passe par l'authentification.

Or, les procédures d'authentification classiques par identifiant et mot de passe ne suffisent plus. Sur les réseaux locaux comme sur Internet, l'écoute de ligne est l'attaque numéro un. L'écoute de ligne permet de récupérer facilement et pratiquement sans risque de détection l'identifiant et le mot de passe que l'utilisateur envoie au serveur ou bien ses codes d'accès lors d'une connexion légitime. Rien de plus simple ensuite pour l'attaquant que de se connecter à son tour en rejouant les mêmes valeurs et ainsi, de se faire passer pour un utilisateur autorisé. Il s'agit là d'usurpation d'identité. Ceci est amplement faciliter par l'utilisation du même mot de passe (souvent facile à retenir) par les humains pour l'accès à différents applications et services.

La deuxième catégorie d'attaque consiste à espionner, simuler, copier ou voler le moyen d'authentification de l'utilisateur. La troisième concerne la récupération des éléments d'authentification des utilisateurs (*credential*) stockés du côté du serveur d'authentification. La quatrième est l'ingénierie sociale qui vise à tromper la vigilance de l'utilisateur en l'amenant astucieusement à révéler volontairement ses mots de passe, ses codes ou ses secrets, ou bien encore à les deviner. En effet, les utilisateurs choisissent souvent des mots de passe faibles (courts, simples, classiques) ou qui leur correspondent (prénom des enfants, dates de naissance, nom du chien de la maison, nom de l'artiste ou du sportif préféré...) afin de les retenir plus facilement. . La cinquième est les attaques dites à base de dictionnaire, à base de tables arc en ciel ou encore celles dites « à force brute ». Ce dernier type consiste par exemple à essayer systématiquement et automatiquement tous les mots de passe possibles ou toutes les clés de chiffrement jusqu'à trouver les bons. Comme les mots de passe utilisés sont souvent

courts (moins de 8 caractères) et simples (lettres et chiffres) l'attaque à force brute est parfois très efficace.

Enfin, il existe d'autres types d'attaques utilisables dans des contextes particuliers comme l'utilisation de programmes malveillants notamment les keylogger, les chevaux de Troie, les logiciels espions ou encore les techniques d'hameçonnage, les attaques à clair connu ou à clair choisi, le fouillage des poubelles, etc.

L'enjeu est d'autant plus considérable que ces menaces qui pèsent sur les particuliers, les entreprises, les organisations, les administrations et leur système d'information sont bien réelles. Elles sont aussi lourdes de conséquences en cas de concrétisation, c'est-à-dire d'attaque réussie par intrusion. Une intrusion frauduleuse dans un système d'information par absence de contrôle des utilisateurs ou par usurpation de l'identité d'un utilisateur autorisé peut avoir des conséquences graves, à la hauteur des droits d'accès et d'action alloués à cet utilisateur.

L'authentification n'est donc pas une fonction de sécurité à négliger, bien au contraire. Elle occupe une place centrale dans la sécurité des réseaux d'aujourd'hui.

L'objectif de ce chapitre est de mettre en avant les mécanismes d'authentification, ainsi que les mécanismes cryptographiques de base. Il sera ainsi organisé comme suit : dans un premier temps nous définissons l'opération d'authentification et ses facteurs. Dans un deuxième temps nous présentons les principes cryptographiques. Dans un troisième et dernier temps nous abordons la classification des méthodes d'authentification, dans le but de situer le mécanisme d'authentification utilisé en IMS : IMS-AKA et ainsi pouvoir comprendre son analyse lors du chapitre 4.

3.1. Notion d'authentification

3.1.1. Définitions

L'authentification est la fonction de sécurité qui consiste à apporter et à contrôler la preuve de l'identité d'une personne, de l'émetteur d'un message, d'un logiciel, d'un serveur logique ou d'un équipement.

En ce qui concerne l'authentification des personnes, la terminologie est la suivante :

- S’identifier consiste à donner/délivrer son identité.
- Identifier une personne consiste à demander et obtenir son identité.
- S’authentifier consiste à apporter/délivrer la preuve de son identité.
- Authentifier consiste à vérifier l’identité d’une personne en lui demandant une preuve tangible de son identité puis en validant ou en invalidant cette preuve.

Il existe des notions connexes à l’authentification qui doivent être comprises pour ne pas entraîner de confusion.

- L’autorisation correspond à l’allocation des droits d’accès aux ressources du système d’information aux utilisateurs authentifiés au préalable.
- Le contrôle d’accès englobe généralement les fonctions :
  – d’identification ;
  – d’authentification ;
  – d’autorisation ;
  – de journalisation ou comptabilité (*accounting*).
- La signature numérique, quant à elle, est une technique cryptographique qui permet d’assurer la non-répudiation, en plus de l’intégrité d’un message et l’authentification de l’émetteur.

3.1.2. Facteurs d’authentification

La façon de s’authentifier est classiquement divisée en trois catégories selon les facteurs employés :

- quelque chose que vous savez : par exemple, un mot de passe, une expression ou un numéro d'identification personnel (PIN), la réponse à un défi (l'utilisateur doit répondre à une question).
- quelque chose que vous possédez : par exemple, carte d'identité, jeton de sécurité, jeton de logiciel, téléphone ou téléphone cellulaire.
- quelque chose que vous êtes : quelque chose lié intrinsèquement à l’utilisateur de de manière unique (par exemple, des empreintes digitales, empreintes rétiniennes, le visage, la voix, réseau veineux ou tout autre identifiant biométrique).

On peut ajouter un quatrième facteur :

- quelque chose que vous savez faire : par exemple une signature manuscrite, lire un mot déformé.

Mais cette quatrième catégorie n'est souvent prise en compte que pour différencier les humains des ordinateurs (les fameux champs « captcha », où l'on vous demande de recopier un mot).

La combinaison de deux facteurs au minimum est l'une des conditions d'une authentification forte. L'authentification sera réputée forte s'il est difficile d'usurper l'identité d'un utilisateur autorisé à qui l'on a fourni au préalable un moyen d'authentification. S'il est relativement facile pour un attaquant de contourner, de tromper ou de casser le procédé d'authentification, comme c'est le cas pour les couples identifiant-mot de passe, on parlera plutôt d'authentification faible.

3.2. Principes cryptographiques

Afin d'effectuer une authentification, il faut s'appuyer sur des mécanismes de cryptographie. La cryptographie est l'outil de base de l'authentification. Elle est classiquement divisée en cryptographie à clés symétriques et clés asymétriques :

3.2.1. La cryptographie symétrique

Une clé symétrique est une information secrète. La cryptographie utilise souvent la cryptographie à clé identique pour à la fois le chiffrement de texte en clair et le déchiffrement de texte chiffré. L'expéditeur et le destinataire doivent convenir d'une méthode pour chiffrer / déchiffrer un message. La clé, en pratique, représente le secret partagé entre deux ou plusieurs parties. Par exemple, un expéditeur effectue son codage par un décalage à droite d'une lettre de l'information qu'il veut transmettre (Bonjour → Cpohpvs) et le récepteur effectue le décodage en décalant d'une lettre à gauche l'information reçue (Cpohpvs → Bonjour) pour déchiffrer le message. Ce secret (décalage à gauche) est appelé clé symétrique. La clé symétrique de chiffrement est également dénommée clé de session, clé secrète ou encore clé partagée.

Des algorithmes pour le chiffrement utilisant des clés symétriques sont par exemple l'algorithme historique, Data Encryption Standard (DES), qui utilisait des clés de 56 bits,

l'algorithme 3DES (avec 3 chiffrements et plusieurs clés) standardisé pour la sécurisation des transferts PPP (RFC 2420).

La cryptographie à clé symétrique avec une longueur de clé appropriée est considérée comme assez forte pour protéger le message. Elle est largement utilisée dans l'Internet. Cependant, le problème de cette solution est la gestion de la clé : comment le récepteur a-t-il connaissance de la clé? Si celle-ci lui est transmise comment s'assurer de sa validité et de sa pérennité.

3.2.2. La cryptographie asymétrique

Elle utilise des clés différentes pour le chiffrement et le déchiffrement. Une de ces deux clés doit être maintenue privée, elle est appelée clé privée, alors que l'autre peut être rendue publique (sa transmission au récepteur n'a pas à être sécurisée), d'où son appellation de clé publique. C'est aussi pourquoi ce type de cryptographie est également nommé cryptographie à clé publique. Un exemple de clé publique est l'identité de l'utilisateur, on parle alors de mécanisme à base d'identité (*Identity Based*).

Un exemple d'algorithme très utilisé pour chiffrer/déchiffrer les messages est RSA (du nom de ses concepteurs : Ron Rivest, Adi Shamir et Leonard Adleman,) qui repose sur le choix aléatoire de nombres premiers par les clients pour générer leurs clés (sa robustesse est liée à la difficulté de la factorisation en nombres premiers).

L'algorithme RSA s'appuie sur des longueurs de clés importantes (au moins 1024), ces tailles peuvent être réduites, au prix d'une complexité de calcul accrue, par l'utilisation de fonctions elliptiques.

Ainsi, contrairement à la cryptographie à clé symétrique, la cryptographie à clé asymétrique ne nécessite pas d'échange sécurisé initial d'une, ou plusieurs clés secrètes entre un émetteur et un récepteur. Les algorithmes de codage/décodage fonctionnent de telle sorte que, tandis qu'il est facile pour le destinataire de générer les clés publiques et privées et de déchiffrer le message en utilisant la clé privée, ainsi que pour l'émetteur de chiffrer le message en utilisant la clé publique, il est extrêmement difficile pour quiconque de deviner la clé privée en fonction d'une connaissance de la clé publique. Les algorithmes s'appuient sur des relations mathématiques qui n'ont pas de solution efficace.

Cependant, le chiffrement à clé asymétrique est considéré comme plus lent que le chiffrement à clé symétrique, c'est pourquoi il n'est généralement utilisé que pour les échanges à haute sécurité tels les échanges de clés session.

3.2.3. Notion de certificat

Les algorithmes de chiffrement asymétrique sont basés sur le partage entre les différents utilisateurs d'une clé publique. Généralement le partage de cette clé se fait au travers d'un annuaire électronique (LDAP par exemple) ou bien d'un site web.

Toutefois ce mode de partage a une grande lacune : rien ne garantit que la clé est bien celle de l'utilisateur à qui elle est associée. En effet un attaquant peut corrompre la clé publique présente dans l'annuaire en la remplaçant par sa clé publique. Ainsi, l'attaquant sera en mesure de déchiffrer tous les messages ayant été chiffrés avec la clé présente dans l'annuaire.

Le certificat permet d'associer une clé publique à une entité (une personne, une machine, ...) afin d'en assurer la validité. Le certificat est en quelque sorte la carte d'identité de la clé publique, délivré par un organisme appelé autorité de certification (souvent notée CA pour Certification Authority).

L'autorité de certification est chargée de délivrer les certificats, de leur assigner une date de validité (équivalent à la date limite de péremption des produits alimentaires), ainsi que de révoquer éventuellement des certificats avant cette date en cas de compromission de la clé (ou du propriétaire).

3.2.4. Structure d'un certificat

Les certificats sont des petits fichiers divisés en deux parties :

- La partie contenant les informations
- La partie contenant la signature de l'autorité de certification

La structure des certificats est normalisée par le standard **X.509** de l'UIT (plus exactement X.509v3), qui définit les informations contenues dans le certificat :

- La version de X.509 à laquelle le certificat correspond;
- Le numéro de série du certificat ;

- L'algorithme de chiffrement utilisé pour signer le certificat ;
- Le nom (DN, pour *Distinguished Name*) de l'autorité de certification émettrice ;
- La date de début de validité du certificat ;
- La date de fin de validité du certificat ;
- L'objet de l'utilisation de la clé publique ;
- La clé publique du propriétaire du certificat ;
- La signature de l'émetteur du certificat (*thumbprint*).

L'ensemble de ces informations (informations + clé publique du demandeur) est signé par l'autorité de certification, cela signifie qu'une fonction de hachage crée une empreinte de ces informations, puis ce condensé est chiffré à l'aide de la clé privée de l'autorité de certification, la clé publique ayant été préalablement largement diffusée afin de permettre aux utilisateurs de vérifier la signature avec la clé publique de l'autorité de certification (Figure 3.1).

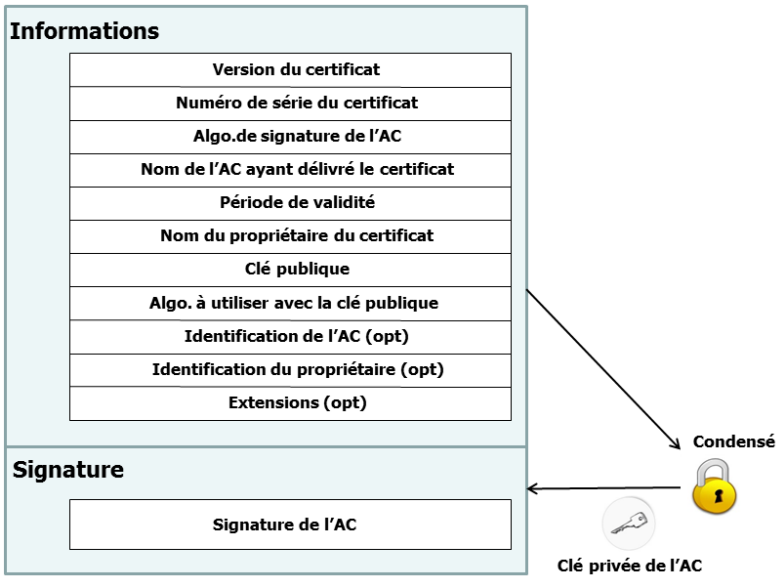


Figure 3.1. Format d'un certificat.

Lorsqu'un utilisateur désire communiquer avec une autre personne, il lui suffit de se procurer le certificat du destinataire. Ce certificat contient le nom du destinataire, ainsi que sa clé publique et est signé par l'autorité de certification. Il est donc possible de vérifier la validité du message en appliquant d'une part la fonction de hachage aux informations contenues dans le certificat, en déchiffrant d'autre part la signature de l'autorité de certification avec la clé publique de cette dernière et en comparant ces deux résultats (Figure 3.2).

3.2.5. Signatures de certificats

On distingue différents types de certificats selon le niveau de signature :

- Les **certificats auto-signés** sont des certificats à usage interne. Signés par un serveur local, ce type de certificat permet de garantir la confidentialité des échanges au sein d'une organisation, par exemple pour le besoin d'un intranet. Il est ainsi possible d'effectuer une authentification des utilisateurs grâce à des certificats auto-signés.
- Les **certificats signés par un organisme de certification** sont nécessaires lorsqu'il s'agit d'assurer la sécurité des échanges avec des utilisateurs anonymes, par exemple dans le cas d'un site web sécurisé accessible au grand public. Le certificateur tiers permet d'assurer à l'utilisateur que le certificat appartient bien à l'organisation à laquelle il est déclaré appartenir.

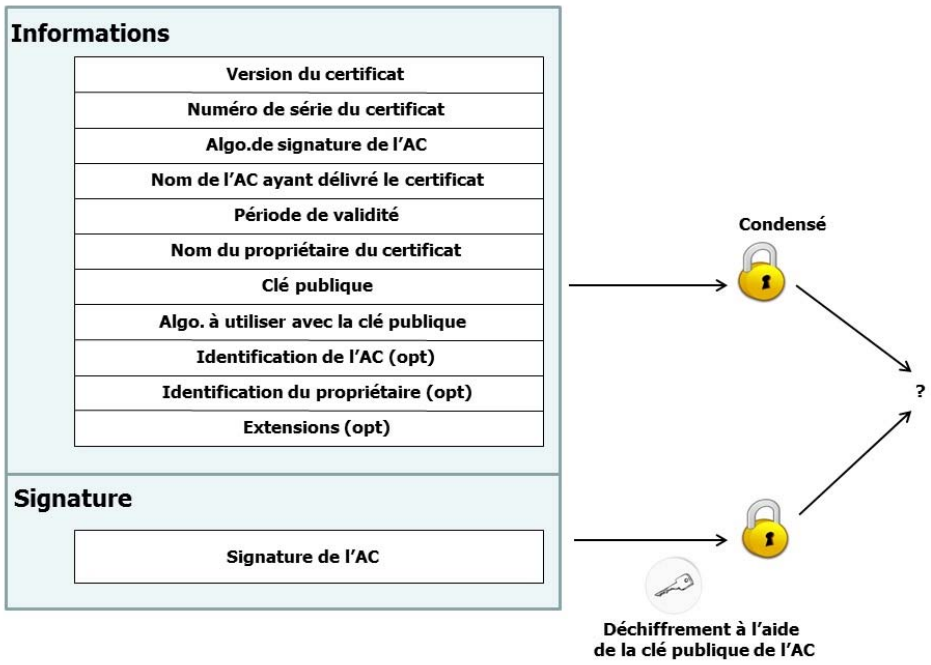


Figure 3.2. Vérification d'un certificat

3.2.6. Exemple d'utilisation conjointe de mécanismes d'authentification : IPsec

Les trois mécanismes élémentaires que nous venons de citer sont généralement associés. Ainsi, la cryptographie par clé asymétrique est utilisable pour négocier des clés

secrètes partagées dans un schéma par clé symétrique après avoir vérifié la validité du certificat de clé publique.

IPsec [KeSe05b] est un exemple de solution standardisée par l'IETF dans Internet et utilisée pour IMS qui met en œuvre plusieurs mécanismes. IPsec permet de transporter des données sécurisées sur Internet en établissant une connexion sécurisée. Pour ce faire une phase d'authentification des extrémités de la connexion a lieu avant de pouvoir échanger des clés puis de transférer de façon protégée les données. Les mécanismes de sécurité : clés de session, certificats, secret partagé sont négociables par protocole.

3.3. Classification des méthodes d'authentification

Voici une classification des méthodes d'authentifications suivant le critère de la sécurité :

3.3.1. Authentification par mots de passe statiques

Cette catégorie est la plus largement utilisée à l'heure actuelle. Elle ne permet pas d'assurer une authentification forte des utilisateurs. Un mot de passe est dit statique (en opposition à dynamique) lorsqu'il ne change pas d'une transaction à l'autre. C'est le cas de la plupart des mots de passe que nous utilisons quotidiennement. Nous le mémorisons et renseignons le champ mot de passe avec sa même valeur chaque fois qu'il nous est demandé.

L'écoute de ligne est l'une des attaques les plus efficaces aussi bien sur les réseaux locaux d'entreprise que sur Internet. De plus, elle est très difficile à déceler. La faiblesse des mots de passe utilisés est l'autre problème majeur : les bons mots de passe sont très difficiles à mémoriser par les utilisateurs donc peu utilisés en pratique.

3.3.2. Authentification par procédés biométriques

Parmi les quatre facteurs d'authentification, les caractéristiques physiologiques de l'individu (quelque chose que vous êtes) est sans doute le facteur le plus populaire dans l'imaginaire des gens. Les capteurs d'empreintes digitales, les scanneurs de forme palmaire et autres lecteurs oculaires sont la panacée des contrôles d'accès sécurisé de nos films d'espionnage préférés. Le fondement scientifique en ce domaine est toutefois irréfutable, car

chaque individu est physiquement différent des autres, même si seulement 1 % de nos chromosomes sont à l'origine de cette différence.

La biométrie est la mesure biologique des individus en ce qu'ils ont de singulier. Cela fait de la valeur d'une mesure de la caractéristique physique d'une personne un élément d'authentification à la fois irréfutable et pratique, toujours disponible. Pourtant, nous verrons que la biométrie semble mieux adaptée à un contrôle d'accès physique à des locaux ou à des équipements sécurisés qu'à un usage généralisé d'authentification des utilisateurs sur Internet. Elle peut toutefois être un bon complément pour contrôler l'accès à un authentifieur, préalablement à l'exécution du processus d'authentification proprement dit.

3.3.3. Authentification par mots de passe dynamiques

Un mot de passe est dit dynamique (en opposition à statique) lorsqu'il change d'une transaction à l'autre. La sécurité est alors renforcée de façon significative, car l'attaque courante par écoute de ligne, puis rejeu du mot de passe récupéré, devient sans objet. Les mots de passe dynamiques sont aussi appelés mots de passe à usage unique ou OTP (One-Time Password).

3.3.4. Protocole défi-réponse

Le protocole défi-réponse, appelé aussi "challenge-réponse" reste le plus sûr des protocoles en matière d'authentification. Pour éviter tout rejeu et ainsi garantir la fraîcheur de l'authentification, on utilise un challenge aléatoire appelé également un nonce ('*number used once*'). Bob envoie le challenge à Alice, Alice doit être la seule à pouvoir fournir la bonne réponse. Le challenge est choisi afin d'empêcher toute attaque par rejeu.

Pour garantir que le défi à chiffré ou à déchiffrer est différent pour chaque exécution du protocole d'authentification, trois techniques sont utilisées. Le défi peut être dérivé soit à partir d'une horloge de lecture en temps réel, dans ce cas il est appelé un horodatage (*timestamps*), à partir d'un compteur qui est incrémenté pour chaque exécution de protocole, dans ce cas il est appelé un(e) compteur/séquence, ou à partir d'un générateur de nombres aléatoires, dans ce cas il est appelé un "nonce/rand". Dans tous les cas, un nouveau défi (Horodatage, séquence ou nonce) est généré pour chaque exécution de protocole.

Avec l'horodatage, un utilisateur A qui veut s'authentifier, chiffre la valeur actuelle de son horloge et envoie le résultat à la partie B demandant l'authentification. Ensuite B déchiffre le message reçu et vérifie que l'horodatage correspond au temps réel courant. L'inconvénient de l'horodatage apparaît donc immédiatement. A et B doivent avoir des horloges synchronisées en temps réel pour que la vérification soit possible. Tandis que les horloges ne peuvent jamais être parfaitement synchronisés et les messages prennent du temps pour être transmit via les réseaux, en tout cas, B ne peut jamais s'attendre à ce que l'horodatage déchiffré reçu à partir de A soit égale à sa propre horloge en temps réel de lecture. L'horodatage de A au mieux peut (et devrait normalement) être dans une certaine fenêtre de temps limitée de l'horloge en temps réel de B. Cependant, dès que la fenêtre de temps de tolérance est définie, un intrus potentiel pourrait l'exploiter à usurper l'identité «A» en rejouant un de ces récent messages d'authentification à l'intérieur de cette fenêtre de temps. Empêcher cela nécessite de limiter le nombre d'exécutions du protocole d'authentification au cours de la fenêtre temporaire, et de sauvegarder, côté B, tous les messages d'authentification échangés à l'intérieur de la fenêtre. Ainsi, l'efficacité et la sécurité exigent une bonne synchronisation de l'horloge. Atteindre une telle synchronisation est souvent difficile. En plus la phase de synchronisation nécessite d'autre mécanismes d'authentification qui ne dépendant pas du temps pour la sécuriser.

Avec la méthode utilisant les compteurs, A et B maintiennent un compteur synchronisé. Ce dernier représente le nombre de fois où les deux entités ont était authentifiées mutuellement. Chaque fois que A veut communiquer avec B, il chiffre son compteur et envoie le résultat à B, qui déchiffre et vérifie qu'il correspond à sa propre valeur de compteur, après quoi les deux parties incrémentent leurs compteurs respectifs pour une future authentification. L'inconvénient de cette méthode est que les deux parties doivent maintenir leurs compteurs synchronisés, ce qui pose des problèmes de stockage et de gestion de compteur en long terme. Les compteurs doivent être suffisamment longs pour empêcher un attaquant d'attendre le point du retour à 0 (*counter wraparound*). L'utilisation des compteurs complique la résolution des conflits lorsque les deux parties veulent initier l'exécution du protocole en même temps. Une fois la perte de synchronisation entre les deux compteurs est détectée, une autre méthode d'authentification doit être utilisée en toute sécurité pour les resynchroniser, par contre il peut y avoir des effets catastrophiques dans le cas d'une une erreur non détectée dans les valeurs des compteurs.

Par conséquent, les deux techniques d'horodatages et de comptage, tout en étant utiles, ne sont pas une solution complète, en particulier pour des protocoles dans les couches inférieures d'une architecture de réseau. La meilleure technique à cet effet, et aussi le plus simple à mettre en œuvre, consiste à utiliser des valeurs à usage unique (nonce).

Le prix de cette simplicité est un message supplémentaire dans le réseau. Alors que A peut s'authentifier auprès de B avec un seul message si la technique d'horodateurs ou des compteurs est utilisée, deux messages sont nécessaires avec des nonces. En effet, c'est l'entité B et non pas A qui doit être sûre que n'a jamais été utilisé auparavant ; c'est donc B qui doit le générer. Ainsi, B doit chiffrer le nonce et l'envoyer à A. A doit le déchiffrer et le renvoyer en clair à B pour l'authentifier, ce qui coûte un total de deux messages.

3.4. Conclusion

Dans ce chapitre nous avons présenté les différents mécanismes d'authentification ainsi que les principes cryptographiques de base. Cette présentation non exhaustive est nécessaire pour bien suivre et comprendre le développement du protocole d'authentification en IMS proposé dans le chapitre suivant. En effet, au niveau du réseau IMS c'est le protocole défi-réponse (avec les séquences et les nonces) qui a été choisi pour l'authentification.

Pour qu'un utilisateur puisse utiliser les services du domaine IMS, tels qu'établir une session multimédia ou recevoir une demande de session, il doit s'enregistrer au réseau IMS. Cet enregistrement nécessite deux authentifications, une authentification au niveau du domaine commutation du paquet (GPRS, WIFI,… par exemple), et une autre authentification au niveau IMS. Partant de cette observation le chapitre suivant présente notre première contribution, le protocole d'authentification *one-way IMS-AKA*.

Chapitre 4: The PKI-based One-way IMS-AKA

L'architecture IMS devient de plus en plus la solution préférée des fournisseurs de services multimédias que ce soit fixes ou mobiles, mais aussi l'une des cibles favorites des attaquants. En effet, la sécurité est un problème capital et qui pose des problèmes qui ne sont pas toujours simples à résoudre, bien que trop souvent délaissé pour diminuer les coûts d'investissement.

Dans les réseaux de téléphonie classiques, le système est quasiment fermé. Auparavant, les commutateurs ne pouvaient être atteints par des informations circulant dans le réseau. Cette particularité provenait de l'unicité de l'application qui circule sur le réseau. Avec le multimédia et l'intégration de la téléphonie dans l'ensemble des données, il devient particulièrement complexe de sécuriser le réseau.

L'un des besoins importants et obligatoires est de garantir une authentification mutuelle entre les utilisateurs et le réseau. Pour ce faire, IMS définit le protocole d'authentification AKA (*Authentication and Key Agreement*) [Gpp09c], qui est basé sur le protocole 3GPP-AKA [Gpp09d], et a un niveau de sécurité similaire. IMS-AKA s'appuie sur le protocole SIP [RSCJ02] et le protocole Diameter [CLGZ03].

Lorsque l'utilisateur veut accéder aux services IMS il doit procéder à deux d'authentifications une au niveau du domaine PS (*Packet-Switch*) en utilisant le protocole 3GPP-AKA, et une deuxième authentification au niveau IMS en utilisant IMS-AKA qui réutilise les mêmes principes de 3GPP-AKA. Avec cette procédure on observera une duplication d'une grande partie des étapes entre les deux authentifications. Malgré cette répétition, les deux authentifications sont nécessaires. Si seulement l'authentification de domaine PS est utilisée un utilisateur malveillant peut usurper l'identité d'autres abonnés IMS en menant une attaque de type utilisation fraudeuse d'IMS (*fraudulent IMS usage*) [LCHW05].

L'objectif de ce chapitre est d'améliorer le protocole IMS-AKA en termes de sécurité, et d'efficacité par la réduction du nombre de messages échangés entre l'utilisateur et le réseau.

Ainsi ce chapitre sera organisé comme suit : Premièrement nous présentons la procédure d'enregistrement dans un réseau IMS. Deuxièmement, nous décrivons quelques travaux connexes. Puis nous enchainons avec le protocole proposé : *PKI-based One-Way IMS-AKA* avant de présenter son implémentation dans la section 4.4. Dans la section 4.5 nous évaluons les performances et la sécurité du protocole proposé, ensuite nous présentons les tests d'évaluations des performances.

4.1. Procédure d'enregistrement dans IMS : Requête REGISTER

Dans cette partie, on va expliquer le déroulement de la procédure d'enregistrement en IMS. C'est une procédure d'accès au réseau IMS qui permet à un terminal de se déclarer joignable de point de vue service IMS. Comme tout autre procédure d'accès (Mise à jour de localisation GSM, attachement GPRS…), le terminal sera authentifié par le réseau IMS et son profil sera chargé dans le S-CSCF nominal qui est une sorte de central de rattachement ou un MSC/VLR qui est alloué à l'utilisateur quelque soit sa localisation dans le monde. Le S-CSCF contient l'adresse du Proxy P-CSCF où le terminal est rattaché.

Il faut garder à l'esprit que le réseau IMS est en dessus de tous types de réseaux qui peuvent servir à l'attachement du terminal au système IMS (GSM, GPRS, UMTS, WiMax, xDSL, RTC…). De plus toutes les procédures d'enregistrement, authentification et chargement de profiles qui se font au niveau IMS sont indépendantes des procédures dans les réseaux d'accès. La localisation géographique du terminal n'est plus importante car il sera toujours rattaché à son réseau nominal à travers le Proxy du réseau visité.

4.1.1. Prérequis pour obtenir le service IMS

Avant qu'un terminal IMS commence toute opération au niveau du réseau IMS, il y a un certain nombre de prérequis préalables qui doivent être respectées. La figure 4.1 montre une vue de haut niveau des conditions requises.

Premièrement, le fournisseur de services IMS doit autoriser l'utilisateur final d'utiliser le service IMS. Ceci se traduit généralement par un abonnement ou par un contrat signé entre l'opérateur du réseau IMS et l'utilisateur. Ce contrat est similaire à une souscription qui autorise un utilisateur final de recevoir et d'établir des appels téléphoniques sur un réseau sans fil.

Deuxièmement, le terminal IMS a besoin d'avoir accès à un accès IP-CAN (réseau offrant une connectivité IP), comme GPRS (sur les réseaux GSM / UMTS), l'ADSL (*Asymmetric Digital Subscriber Line*) ou WLAN (*Wireless Local Area Network*).

IP-CAN permet d'accéder au réseau IMS, que soit un réseau local ou visité. Dans le cadre de cette condition le terminal IMS doit acquérir une adresse IP (les procédures d'accès GPRS sont décrites dans [Gpp08]). Cette adresse IP est généralement allouée dynamiquement par l'opérateur IP-CAN pendant une période de temps déterminée.

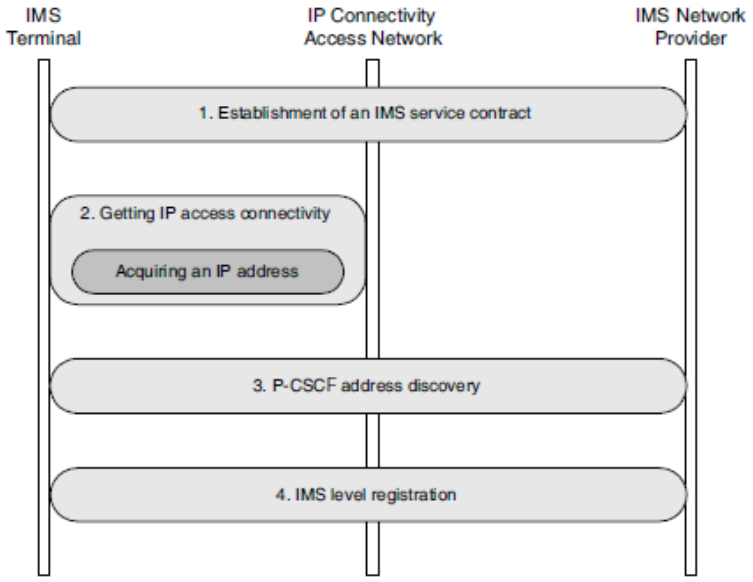


Figure 4.1. Prérequis pour obtenir le service IMS

Lorsque ces deux conditions sont remplies le terminal IMS a besoin de découvrir l'adresse IP du P-CSCF qui agira comme un serveur proxy SIP d'entrée/sortie au réseau IMS pour le terminal. Tous les messages de signalisation SIP envoyés par le terminal IMS traversent ce P-CSCF.

Lorsque la procédure de la découverte du P-CSCF est achevée, le terminal IMS et le P-CSCF sont en mesure d'envoyer et de recevoir des messages de signalisation SIP. Le proxy P-CSCF est affecté en permanence pendant toute la durée de l'enregistrement IMS.

Selon le réseau d'accès IP en cours d'utilisation, la procédure de découverte du P-CSCF peut avoir lieu dans le cadre du processus d'obtention de la connectivité IP-CAN ou comme une procédure distincte. Une procédure séparée est obtenue au moyen de protocole DHCP

(Dynamic Host Configuration Protocol, spécifié dans [Rdro97]) ou DHCPv6 (DHCP pour IPv6, spécifiée dans [Rdro03]).

Lorsque les conditions précédentes sont remplies, le terminal IMS peut commencer l'étape d'enregistrement IMS au niveau SIP.

Afin d'expliquer la procédure, on va prendre l'hypothèse que le terminal est un terminal GPRS qui est dans son réseau nominal et que l'utilisateur s'attache au service IMS de son opérateur. Donc au préalable le mobile a déjà établit un contrat d'accès au service IMS de son opérateur GPRS. (C'est la même procédure si l'utilisateur est dans un réseau visité).

Cette hypothèse reste valable tout au long de ce chapitre. La procédure d'enregistrement se compose des étapes suivantes :

1. Attachement au réseau GPRS.
2. Activation d'un Contexte PDP, avec obtention d'une adresse IPv4/IPv6.
3. Découvert du P-CSCF.
4. Enregistrement IMS.

La figure 4.2 présente les prérequis pour obtenir le service IMS dans ce cas.

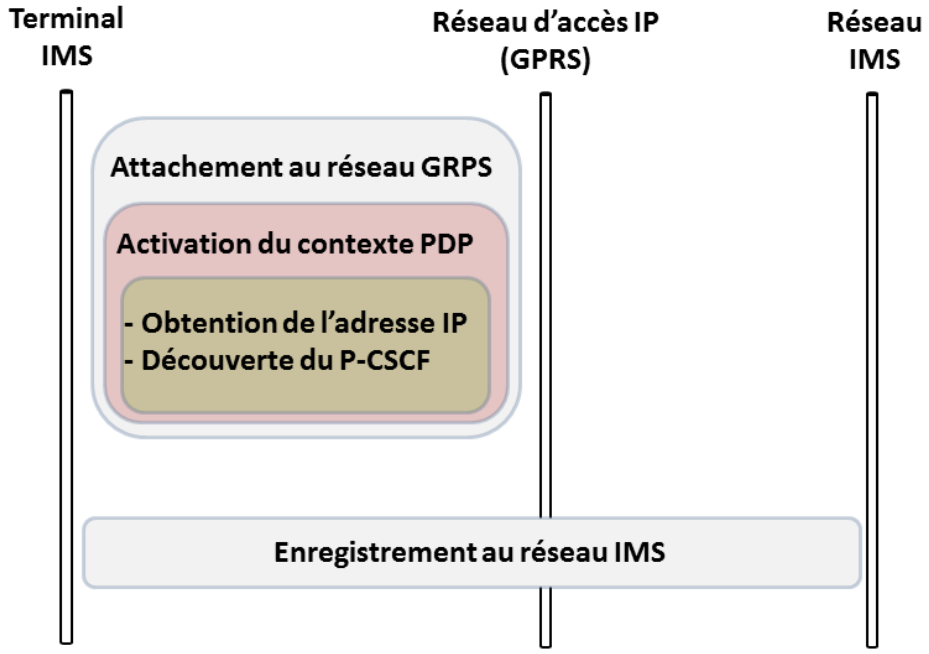


Figure 4.2. Prérequis pour obtenir le service IMS (cas du GPRS comme réseau d'accès)

4.1.2. Connectivité IP en utilisant GPRS

4.1.2.1. Déroulement global :

En GPRS, le terminal IMS initie d'abord un ensemble de procédures, connues sous le nom du *GPRS attach*. Ces procédures, qui seront expliquées dans la section suivante, impliquent plusieurs nœuds, allant du SGSN (*Serving GPRS Support Node*) au HLR et GGSN. Les procédures sont illustrées à la Figure 4.3. Une fois ces procédures sont terminées le terminal envoie une activation de contexte PDP message de demande au SGSN demandant une connexion à un réseau IPv4 ou IPv6.

Le GGSN va fournir au mobile un préfixe d'adresse IPv6 de 64bits (au lieu d'une adresse complète) et l'envoie dans la réponse à l'ouverture du PDP contexte. Le SGSN transmet d'une façon transparente le préfixe IPv6 au terminal qui lui va choisir aléatoirement un suffixe IPv6 de 64bits, pour former en tout, une adresse IPv6 de 128bits. Si le terminal a demandé une connexion IPv4, le GGSN fournit le terminal une Adresse IPv4.

Notons que si l'IP CAN n'est pas du type GPRS/UMTS, le terminal obtiendra une adresse IPv6 en utilisant probablement un protocole tel que le DHCPv6.

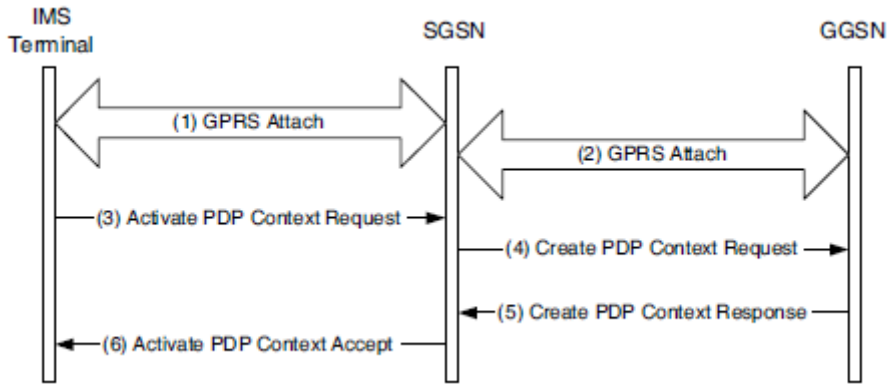


Figure 4.3. Obtention de la connectivité IP en GPRS

4.1.2.2. Attachement au réseau GPRS :

Quand une station mobile (MS pour Mobile Station), demande accès au réseau GPRS, la MS envoie une demande d'attachement au SGSN. Ce message va déclencher l'authentification GPRS, qui est mis en œuvre par le module de gestion de la mobilité GPRS (GMM pour GPRS Mobility Management) et le module MAP (Mobile Application Part) du

système de signalisation 7 (SS7), entre MS et SGSN et entre SGSN et HSS/AUC respectivement. Cette procédure se compose des étapes suivantes (voir Figure 4.4).

G1 : Considérons une MS avec un imsi comme valeur de son IMSI (*International Mobile Subscriber Identity*). Pour accéder aux services GPRS, la MS envoie une demande d'attachement GMM (*GPRS Mobility Management*) avec le paramètre IMSI = imsi au SGSN.

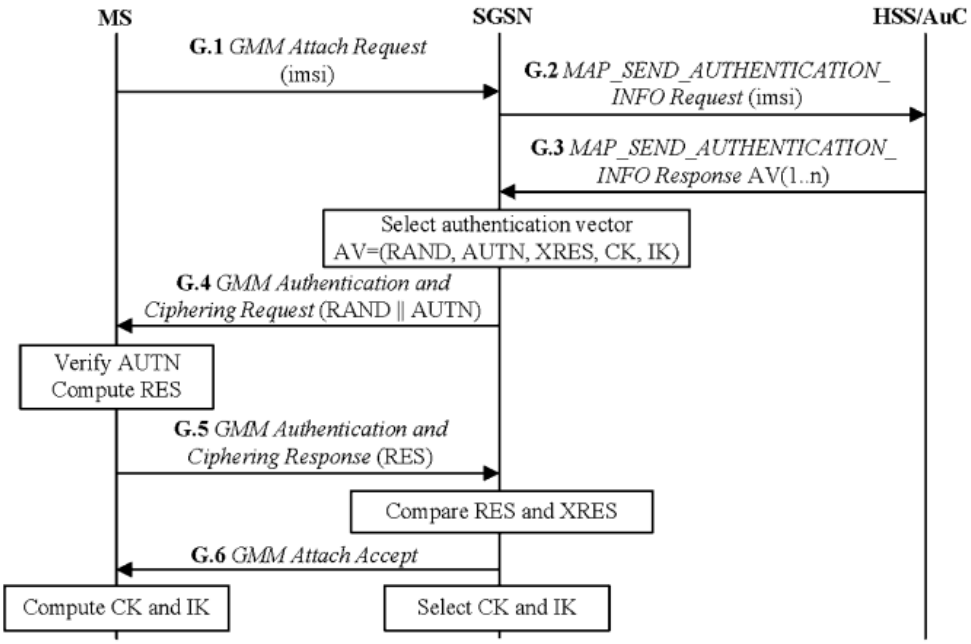


Figure 4.4. Le flux de messages pour l'authentification 3GPP (niveau GPRS)

G2 : Si SGSN dispose des informations propres à l'utilisateur, qui vont permettre de générer un challenge pour authentifier l'utilisateur (ces informations sont appelées vecteurs d'authentification AV), cette étape G2 et l'étape G3 sont ignorés. Sinon, SGSN doit obtenir un AV du HSS/AUC. Autrement dit, SGSN invoque la procédure de distribution des vecteurs d'authentification en envoyant une requête de type MAP SEND AUTHENTICATION INFO au HSS/AUC (avec le paramètre IMSI = imsi).

Un vecteur d'authentification comprend un nombre aléatoire RAND, une réponse attendus depuis l'utilisateur XRES, une clé de chiffrement CK, une clé d'intégrité IK, et un jeton d'authentification AUTN. Chaque AV est utilisé pour une authentification et un accord de clés entre SGSN et MS.

G3 : HSS/AUC utilise la valeur imsi pour récupérer l'enregistrement du MS, et génère un tableau ordonné d'AVs (basé sur la clé secrète K pré-partagée avec MS). Le tableau des AVs généré est envoyé au SGSN à travers une réponse de type MAP SEND AUTHENTICATION INFO.

G4 : SGSN sélectionne le prochain vecteur d'authentification non utilisé dans le tableau ordonnée des Avs et envoie les paramètres RAND et AUTN à la station mobile par le biais d'une requête *GMM Authentication and Ciphering*.

G5 : MS vérifie si AUTN reçu est valide. Si oui, il produit une réponse RES qui est renvoyé au SGSN (Réponse *GMM Authentication and Ciphering*). Le SGSN compare la RES reçus avec la XRES. Si elles sont égales, alors l'authentification et l'accord sur les clés sont terminés avec succès.

G6 : Le SGSN envoie un message *GMM Attach Accept*, la procédure est ainsi terminée.

Après l'authentification GPRS, la MS procède à l'activation de son PDP (*Paquet Data Protocol*) pour obtenir l'accès au réseau GPRS. Le contexte PDP spécifie les protocoles applicatifs ainsi que les informations de routage utilisées pour la session de communication GPRS.

4.1.3. Découvert du P-CSCF

Il y a deux façons pour obtenir l'adresse IP de P-CSCF :

- Intégré (*Integrated*) dans la procédure d'accès à l'IP CAN, lors de l'établissement du contexte PDP le terminal obtiendra non seulement une adresse IPv4/IPv6 mais aussi l'adresse du P-CSCF.
- La *stand-alone*, dans laquelle la découverte du P-CSCF se fait grâce à l'utilisation du DHCP/DHCPv6 et du DNS.

Une fois un P-CSCF est alloué à un utilisateur il le sera toujours jusqu'à la prochaine découverte de P-CSCF. Et le terminal IMS n'a pas à s'inquiété si l'adresse du P-CSCF a changé car elle est fixe

4.1.4. Enregistrement et authentification IMS

Une fois le terminal IMS a suivi les procédures d'obtenir une connectivité IP au niveau du réseau d'accès, a acquis une adresse IPv4 ou construit une adresse IPv6, et a découvert l'adresse IPv4 ou IPv6 de son P-CSCF, le terminal IMS peut commencer l'enregistrement au niveau IMS.

L'enregistrement d'un utilisateur dans le réseau est la première action réalisée par un terminal, dès sa mise en route. Elle est indispensable puisqu'elle permet à la fois d'appeler et d'être joignable par ses correspondants.

On distingue deux façons pour faire l'enregistrement IMS. La différence réside dans la méthode d'authentification qui est appliqué. Or pour authentifier les utilisateurs le terminal devra être équipé par un UICC, qui peut inclure une application ISIM, USIM ou les deux.

4.1.4.1. Enregistrement IMS avec une ISIM

La procédure d'enregistrement très similaire dans les deux cas même si quelque détail sont différents. On prendra dans un premier temps l'enregistrement utilisant une carte ISIM.

La procédure d'enregistrement permet de réaliser les fonctionnalités suivantes :

- Effectuer l'association entre une "*Public User Identity*" et une adresse IP de contacte.
- Le réseau nominal authentifie l'utilisateur.
- L'utilisateur authentifie son réseau nominal.
- Le réseau nominal IMS autorise l'enregistrement de l'utilisateur et l'utilisation des ressources IMS.
- Si le P-CSCF est localisé dans un réseau visité, le réseau nominal vérifie s'il y a un accord de roaming entre eux et en conséquence il autorise l'utilisation du P-CSCF.
- Le réseau nominal informe l'utilisateur des autres adresses qu'il lui a alloué.
- Le terminal IMS négocie avec le P-CSCF les mécanismes de sécurité à utiliser pour la signalisation qui suit. Ils établissent un ensemble de mécanismes de sécurité pour assurer l'intégrité des messages SIP envoyés.
- Le terminal IMS et le P-CSCF échangent leurs algorithmes de compression des entêtes SIP.

La méthode associée à cette fonctionnalité (enregistrement IMS) est REGISTRER, du protocole de signalisation SIP. Nous allons détailler le processus mis en œuvre avec SIP. Ce processus nommé IMS-AKA peut globalement se diviser comme illustre la figure 4.5 en deux étapes SIP, correspondant à la succession de deux requêtes SIP REGISTER et de leurs réponses.

Première phase

I1 (Messages 1 et 2) : Le terminal IMS (*UE : User Equipment*) envoi un SIP REGISTER message (requête d'enregistrement) contenant son IMPU (*IP Multimedia Public Identity*) et son IMPI (*IP Multimedia Private Identity*) au P-CSCF. Celui-ci ne connaît pas nécessairement le serveur I-CSCF (en général, le P-CSCF appartient au réseau visité). Pour localiser I-CSCF, le serveur P-CSCF procède à une requête DNS à partir du nom de domaine que l'utilisateur a fourni. Une fois localisé, le P-CSCF renvoi la requête au I-CSCF, dans laquelle il a ajouté un champ d'entête *P-VISITED-NETWORK-ID*. Ce champ contient un identifiant du réseau dans lequel le P-CSCF se trouve. Il permettra au S-CSCF de vérifier que le réseau visité bénéficie d'un accord d'itinérance (*roaming*) avec l'opérateur de l'utilisateur. Un autre champ ajouté par le P-CSCF est le champ d'en-tête PATH qui spécifie l'adresse SIP du P-CSCF. Cette information permettra de retourner la réponse à cette requête via ce même serveur P-CSCF.

I2 (Messages 3 et 4) : Lorsque I-CSCF reçoit la requête, il ignore si elle concerne un utilisateur qui est déjà enregistré ou s'il s'agit d'un nouvel enregistrement (c'est un serveur *Stateless*). Le I-CSCF utilise le protocole Diameter via l'interface Cx pour contacter la base de données HSS et lui demander, à partir des identités publiques et privées contenues dans le message de requête REGISTER, d'authentifier l'utilisateur (requête UAR). Si l'utilisateur a déjà été enregistré, un serveur S-CSCF lui a déjà été attribué, et l'adresse de ce serveur est stockée dans la base HSS. Dans ce cas, la base HSS fournit dans sa réponse (message UAA) l'adresse du S-CSCF qui est en charge de la session de l'utilisateur. Autrement, et s'il s'agit d'une première connexion pour l'utilisateur, aucun serveur S-CSCF ne lui a été attribué et la réponse UAA de la base HSS propose l'ensemble des S-CSCF disponibles avec leur caractéristiques propres, de manière que le serveur I-CSCF puisse choisir l'un d'entre eux, selon un critère qu'il détermine librement (généralement pour répartir la charge des utilisateurs équitablement entre tous les serveurs S-CSCF).

I3 (Message 5) : Considérons dans notre scénario qu'il s'agit d'une première authentification : le serveur I-CSCF a donc choisi un serveur S-CSCF pour traiter la session courante de l'utilisateur, et il lui relaie la requête d'enregistrement de l'utilisateur.

I4 (Message 6) : Si S-CSCF n'a pas les informations propres à l'utilisateur, qui vont permettre de générer un challenge pour authentifier l'utilisateur (ces informations sont appelées vecteurs d'authentification AV), le S-CSCF envoie un message MAR (*Multimedia Authentication Request*) via l'interface Cx au HSS pour obtenir le AV, sinon cette étape et l'étape I5 ne sont pas exécutées et on passe directement à l'étape I6. L'AV contient un nombre aléatoire RAND, une réponse attendu XRES (*eXpected RESponse*), une clé de chiffrement CK (*Cipher Key*), une clé d'intégrité IK (*Integrity key*) et un jeton d'authentification AUTN.

I5 (Message 7) : HSS répond par un message MAA (*Multimedia Authentication Answer*) qui, d'une part, confirme l'enregistrement du S-SCSF affecté à l'utilisateur et, d'autre part, retourne une liste ordonnée d'AV propre à l'utilisateur.

I6 (Messages 8 et 9) : Lorsque le S-CSCF les reçoit, il répond au serveur I-CSCF par un message SIP 401 UNAUTHORIZED, contenant le challenge sous la forme d'un champ d'entête appelé WWW-AUTHENTICATE. Ce message de réponse est relayé conformément au modèle SIP client/serveur, c'est-à-dire de proche en proche, en passant par tous les émetteurs de requêtes : le I-CSCF d'abord, le P-CSCF ensuite et le terminal client enfin. Cette liste de serveurs est facilement déterminée car, lors de l'envoi de la requête REGISTER, les serveurs traversés ont enregistré leur adresse SIP, grâce au champ d'en-tête PATH.

I7 (Message 10) : Avant que P-CSCF transfert le message *SIP 401 Unauthorized* au terminal client il garde les deux clés CK et IK et donc le message envoyé ne contient que les paramètres RAND et AUTN.

Seconde phase

I8 (Messages 11-15) : Lorsque le UE reçoit le message de réponse 401, il vérifie le challenge AUTN. Si le résultat est positif, donc le S-CSCF (réseau nominal) est bien authentifié. Par la suite, UE calcule la réponse (RES) à envoyer au S-CSCF, il calcule aussi les deux clés CK et IK.

Cette réponse est générée dans une nouvelle requête d'enregistrement REGISTER. Elle est envoyée en suivant le même processus d'acheminement que le premier message de requête : le terminal client ne connaît que le serveur P-CSCF, qui, lui-même, ne peut s'adresser qu'au I-CSCF, qui, à son tour, sous-traite la demande auprès du serveur S-CSCF.

I9 (Message 16) : À réception de la requête, le serveur S-CSCF vérifie l'authentification de l'utilisateur en reprenant les vecteurs d'authentification que lui a fournis le HSS à l'étape précédente et qu'il a stockés. Si les paramètres d'authentification sont valides, c.-à-d. si le XRES est égale à RES, l'authentification est un succès, et le serveur S-CSCF en informe le HSS (par un message SAR : *Server Assignment Request*). Dès lors, l'utilisateur peut appeler et devient joignable à l'adresse qui figure dans le champ d'en-tête CONTACT de la requête REGISTER, et ce pendant toute la durée mentionnée dans la valeur de l'en-tête EXPIRES.

I10 (Message 17) : Le HSS répond ensuite au S-CSCF en lui envoyant le profil complet de l'utilisateur, qui est stocké temporairement et servira à paramétrer et personnaliser les services de ce dernier.

I11 (Messages 18-20) : Pour terminer, le serveur S-CSCF envoie un message de réponse 200 OK au UE pour lui notifier le succès de l'opération d'enregistrement. Ce message comporte notamment un champ *P-ASSOCIATED-URI*, qui liste les adresses SIP que l'utilisateur a enregistrées auprès du HSS (et donc avec lesquelles il est joignable), et un champ *SERVICE-ROUTE*, qui précise explicitement l'adresse URI SIP du serveur S-CSCF que l'utilisateur pourra contacter directement dans ses requêtes suivantes (bien qu'elles transiteront obligatoirement avant par le serveur P-CSCF).

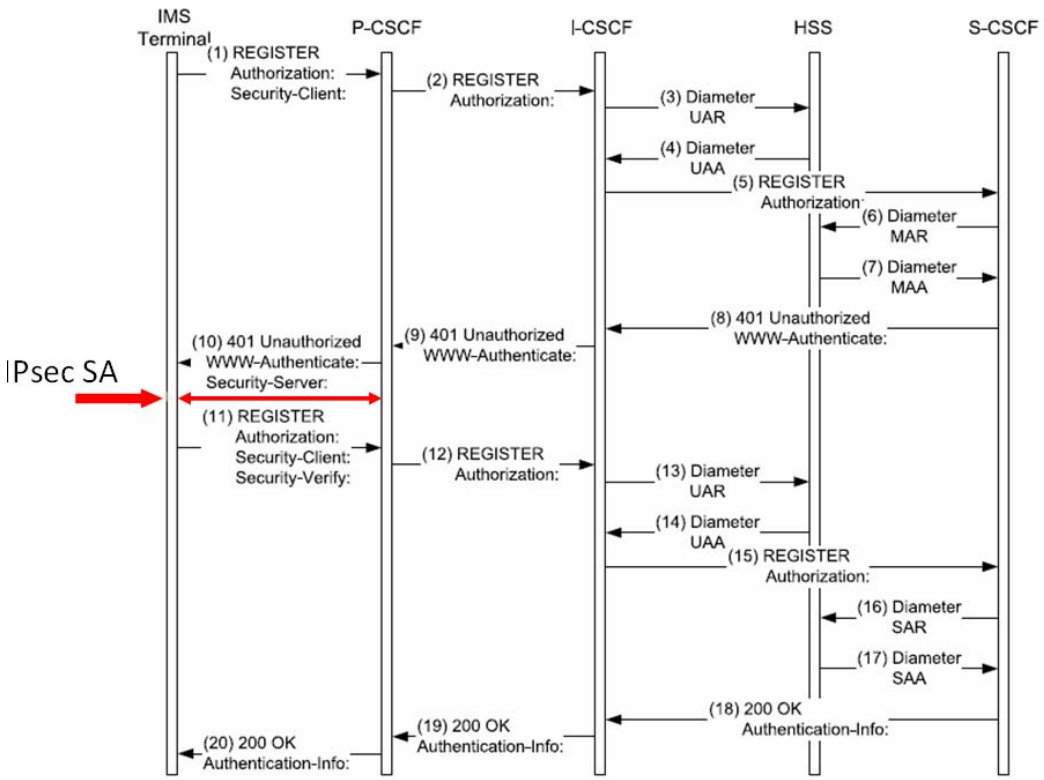


Figure 4.5. Processus d'authentification/enregistrement en IMS avec le protocole IMS-AKA

Analyse :

1. Dans la procédure ci-dessus, les étapes I1-I8 exécutent l'authentification, et les étapes I9-I11 effectuent l'enregistrement.
2. Dans la seconde phase du processus, les messages entre les UE et P-CSCF sont protégés par une association de sécurité IPSec. Cette association est négociée au cours de la première phase à travers les deux champs "Security-Client" (message 1, figure 4.5) et "*Security-Server*" (message 10).
3. Le champ "*Security-Verify*", qui n'est d'autre que l'écho du champ "Security-Server", est ajouté par le terminal pour contrer les attaques de types man-in-the midle. En effet, avec l'ajout du Security-Verify, un attaquant qui a modifié la liste mentionnée dans le "Security-Server" doit casser le mécanisme de sécurité choisi en temps réel pour modifier le champ "Security-Verify". Sinon, le P-CSCF remarque l'attaque et annule l'enregistrement. Cette façon de négociation est bien sécurisée tant que le mécanisme le plus faible dans la liste ne peut être brisé en temps réel.

La Figure 4.6 donne un exemple complet de message SIP (message 11 en figure 4.5), on peut voir clairement le champ *Security-Server*.

```
REGISTER sip:home1.net SIP/2.0
Via: SIP/2.0/UDP [1080::8:800:200C:417A]:5059;comp=sigcomp;
                 branch=z9hG4bK9h9ab
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD;
                       utran-cell-id-3gpp=24450289A3299239
From: <sip:alice@home1.net>;tag=s8732n
To: <sip:alice@home1.net>
Contact: <sip:[1080::8:800:200C:417A]:5059;comp=sigcomp>
         ;expires=600000
Call-ID: 23fi57lju
Authorization: Digest username="alice_private@home1.net",
               realm="home1.net",
               nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
               algorithm=AKAv1-MD5,
               uri="sip:home1.net",
               response="6629fae49393a05397450978507c4ef1"
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96;
                 spi-c=909767; spi-s=421909;
                 port-c:4444; port-s=5058
Require: sec-agree
Proxy-Require: sec-agree
Cseq: 2 REGISTER
Supported: path
Content-Length: 0
```

Figure 4.6. Exemple d'une requête SIP REGISTER (Message 11)
[Gonz00]

4.1.4.2. Enregistrement IMS avec une USIM

Si on se place dans un contexte purement UMTS où le terminal n'a pas un ISIM mais plutôt un USIM. Dans ce cas l'utilisateur n'est pas capable d'obtenir une Private User Idententity, une Public User Identity et l'URI du réseau nominal, paramètres nécessaire pour envoyer la requête SIP REGISTER.

Cependant le mobile dispose d'un IMSI, qui est un identifiant international unique pour l'utilisateur. Cet identifiant sera exploité pour que le terminal puisse construire une *Private User Idententity* temporaire, une identité publique temporaire de l'utilisateur (*Public User Identity*) et l'URI du réseau nominal. Ceci va permettre à l'utilisateur de construire une requête SIP REGISTER. Après l'enregistrement l'utilisateur obtiendra des identités publiques (Public User Identities) qu'il utilisera dans les requêtes SIP suivante.

La figure 4.7 illustre la structure d'un IMSI. A partir de la gauche, les trois premiers chiffres constituent le MCC (*Mobile Country Code*). Le MCC représente le pays du réseau nominal. Le MCC est suivi par deux ou trois chiffres qui constituent le MNC (Mobile Network Code). Le MNC représente l'opérateur nominal dans le pays représenté par le MCC. Les autres chiffres constituent le numéro d'identification de l'abonné mobile MSIN (*Mobile Subscriber Identification Number*).

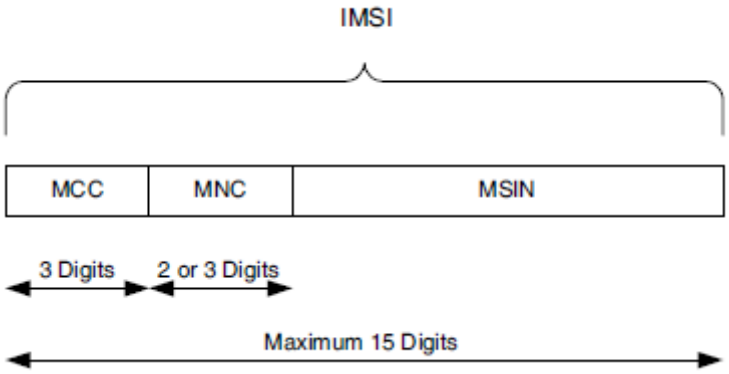


Figure 4.7. Structure du IMSI

- L'identité privée temporaire: elle est toujours du format username@realm. L'IMSI sera le nom de l'utilisateur (username). Le format du domaine (realm) sera de la forme ims.mnc[MNC].mcc[MCC].3gppnetwork.org.
  Supposons qu'on a un IMSI=2483235551234, tel que le MCC=248, le MNC=323 et le MSIN=5551234. Donc l'identité privée temporaire sera : « 2483235551234@ims.mnc323.mcc248.3gppnetwork.org ».
  Notons que l'identité publique temporaire n'est pas une URI SIP.
- Un terminal IMS sans ISIM doit également construire une identité publique temporaire pour s'enregistrer. Une identité publique est une URI SIP qui prend le format sip : username@realm. Il est très simple à construire une identité temporaire publique, car il prend le même format que l'identité privée temporaire, mais maintenant, elle est préfixée par la chaîne : "sip:", puisque l'identité est une URI SIP.
  Donc, si l'on prend comme exemple le même IMSI que nous avons choisi pour illustrer une identité de privée temporaire, l'identité publique temporaire sera de la forme suivante :
  « sip : 2483235551234@ims.mnc323.mcc248.3gppnetwork.org ».

- L'URI du réseau nominal : est obtenu en enlevant la partie utilisateur de l'identité publique temporaire. Selon l'exemple que nous avons suivi l'URI du domaine de réseau nominal est :
  sip : ims.mnc323.mcc248.3gppnetwork.org

La procédure d'enregistrement reste toujours la même, mais le contenu des messages sera différent, puisque les identités, publiques ou privées qui circulent sont temporaires.

4.1.4.3. GPP two-pass authentication

On remarque que pour accéder au réseau IMS, le terminal doit passer par deux authentifications successives, une authentification au niveau du domaine commutation de paquet (GPRS par exemple) en utilisant l'algorithme 3GPP-AKA, et une autre authentification au niveau IMS se basant sur l'algorithme IMS-AKA, qui réutilise les mêmes principes de 3GPP-AKA. Ce qui revient à dire que les deux phases d'authentification se basent sur le même algorithme (3GPP-AKA). Raison pour laquelle elles sont nommées "3GPP two-pass authentication".

Avec cette procédure on observera une duplication d'une grande partie des étapes entre les deux authentifications, le Tableau 4.1 montre les parties dupliquées, les Gi et Ii se réfèrent aux numéros des étapes expliqués en sections 4.1.2.2 et 4.1.4.1.

Malheureusement, malgré cette répétition, les deux authentifications sont nécessaires. En effet, si seulement l'authentification GPRS (domaine commutation de paquet) est utilisée, un utilisateur malveillant peut usurper l'identité des autres abonnés IMS en menant une attaque de type utilisation fraudeuse d'IMS (*fraudulent IMS usage*) [LCHW05].

Tableau 4.1. Etapes identiques entre GPRS et IMS authentification

Authentification GPRS (SS7 MAP)	Authentification IMS (SIP/Cx)
G2. MAP_SEND_AUTHENTIOCATION_INFO Request, **Paramètre :** IMSI	I4. Multimedia Authentication Request **Paramètre** : IMPI
G3. MAP_SEND_AUTHENTICATION_INFO Response,	I5. Multimedia Authentication Answer

Paramètre : AV[1…n]	**Paramètre** : AV[1..n]
G4. User Authentication Request **Paramètre :** RAND || AUTN	I6. 401 Unauthorized **Paramètre** : RAND || AUTN
G5. User Authentication Response **Paramètre :** RES	I8. Register **Paramètre** : RES
G6. GMM Attach Accept	I11. 200 OK

Prenons l'exemple de la Figure 4.8, où il y a deux MSs. MS-A identifié par le couple imsi-A et impi-A dans le réseau GPRS et IMS respectives, et MS-B identifié par imsi-B et impi-B.

Supposons que MS-B est un utilisateur légitime au niveau GPRS et qu'il a réussi l'authentification GPRS (en utilisant imsi-B) pour obtenir l'accès au réseau GPRS. Si aucune authentification IMS n'est nécessaire, MS-B peut effectuer l'enregistrement IMS en envoyant au CSCF une demande d'enregistrement qui comprend l'identité de MS-A (impi-A) comme paramètre. Le CSCF considéra cet enregistrement IMS comme une action légale initiée par MS-A. Par conséquent, MS-B peut accéder illégalement aux services IMS de MS-A.

L'exemple ci-dessus montre que l'authentification IMS-niveau est nécessaire pour empêcher l'accès illégal aux services IMS.

Donc les deux authentifications au niveau GPRS et IMS sont nécessaires, mais le fait qu'IMS-AKA est basé sur l'algorithme 3GPP-AKA implique une duplication de la plupart des étapes de ce dernier.



Figure 4.8. Enregistrement illégal au réseau IMS

Cette procédure d'authentification (GPP two-pass authentication) est généralement utilisée pour les terminaux connectés depuis un réseau d'accès sans fil utilisant une interface air (GPRS, UMTS, WIFI,…). En effet, l'algorithme AKA est utilisé comme algorithme de référence en authentification pour la plupart des réseaux sans fil.

Dans ce contexte l'utilisation de la procédure GPP two-pass authentication est non efficace et a un coût très élevé. Cela est dû aux nombre de messages échangés au niveau de l'interface air.

Pour résoudre ce problème, nous proposons un nouveau mécanisme d'authentification, qui garde les deux authentifications nécessaires (réseau d'accès et IMS), mais qui améliore le protocole IMS-AKA en termes de nombre de message échangés et donc gagner en termes d'efficacité. En plus, notre protocole apporte de nouvelles améliorations à l'IMS-AKA en termes de sécurité aussi.

Dans ce qui suit nous présentons d'abord deux travaux qui ont essayé de résoudre ce problème, cependant ils souffrent de plusieurs failles de sécurité que nous identifions avant de présenter notre protocole.

4.2. Travaux connexes

4.2.1. Approche 1 : One-Pass Authentication

Dans cette approche Lin et al. [LCHW05] propose un mécanisme d'authentification simple qui évite les opérations dupliquées en IMS-AKA et augmente l'efficacité du processus. Pour ce faire, elle s'appuie sur l'IMSI, paramètre que le GPRS stocke pendant l'authentification du terminal, en l'ajoutant au message SIP REGISTER. Le flux de messages est réduit, et l'authentification est réussie si l'IMSI stockées par le HSS est égal à celui envoyé par GPRS. Ceci élimine l'utilisation des vecteurs d'authentification (AV). Dans cette approche, SGSN implémente une passerelle applicative SIP qui modifie le format des messages SIP pour ajouter le paramètre IMSI à la requête SIP REGISTER.

La Figure 4.9 montre le flux des messages d'authentification pour cette approche

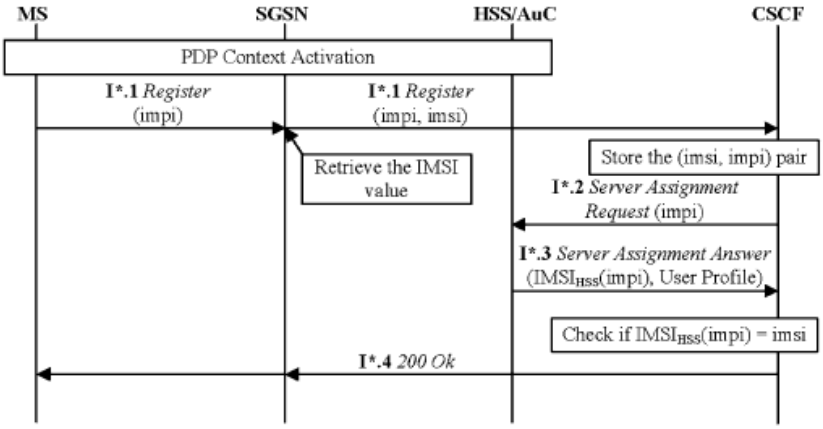


Figure 4.9. Processus du protocole "One-Pass Authentication"

4.2.2. Approche 2 : One-Pass AKA

Cette solution [HuLi07] analyse les problèmes de sécurité existants dans le schéma précédent, et propose une amélioration du processus d'authentification, sans perdre d'efficacité. Le flux de messages est le suivant :

1. Le terminal commence à relever le défi en envoyant un message "digest-reponse" avec un horodatage.
2. Le S-CSCF vérifie le message, s'il est exact il authentifie le terminal et réponds avec un message "reponse-auth" et le vecteur d'authentification (AV).
3. Le terminal vérifie le message "reponse-auth", s'il est valide, il suppose que le S-CSCF est légal. Ensuite le terminal calcule les clés de chiffrement et d'intégrité qui seront partagées avec le P-CSCF pour la protection des données.

La Figure 4.10 montre le flux des messages d'authentification pour cette approche.

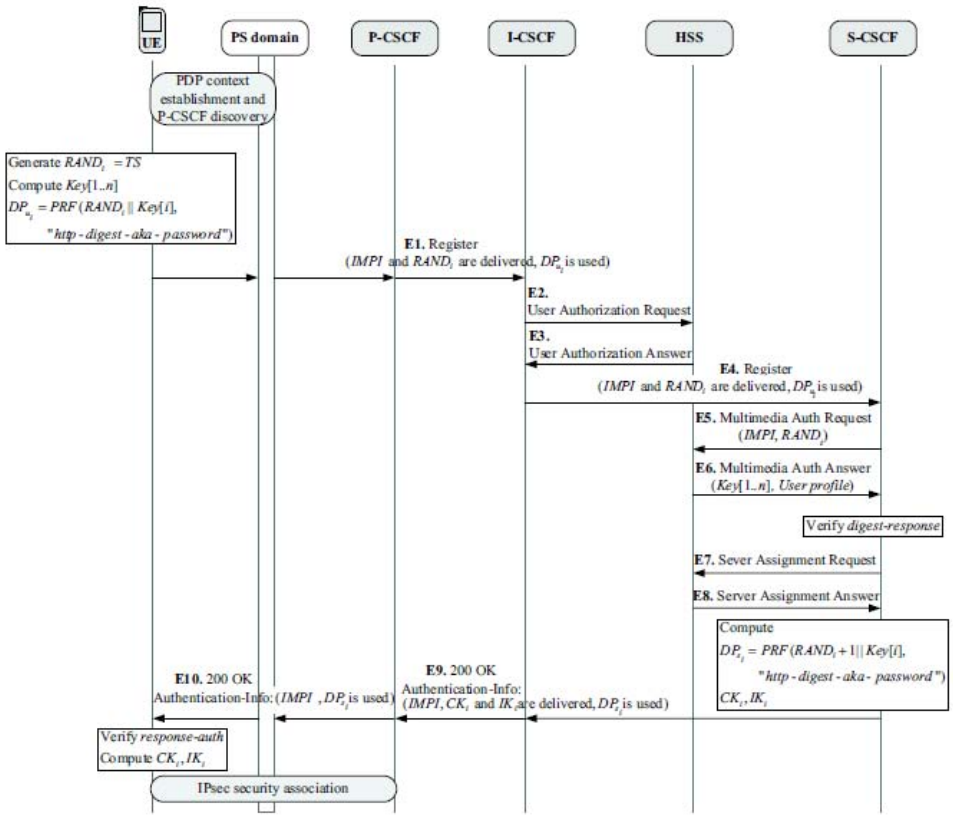


Figure 4.10. Processus du protocole "The Proposed One-Pass AKA"

4.2.3. Analyse de sécurité:

Notre analyse de ces deux schémas nous a permis d'identifier les vulnérabilités suivantes :

- Pas d'authentification mutuelle : dans le premier schéma le terminal n'authentifie pas le S-CSCF (Réseau IMS). En effet, le processus permet juste la vérification du UE par le réseau puisque le S-CSCF vérifie la correspondance entre le IMSI et IMPI du terminal à authentifier.
- Perte de la propriété "key agreement" du protocole IMS-AKA : dans le premier schéma la clé de chiffrement CK et la clé d'intégrité IK ne sont plus utilisées entre le terminal et le P-CSCF pour assurer la confidentialité et l'intégrité des messages échangés entre le terminal et le P-CSCF.
- Pas de garantie d'intégrité et de confidentialité: les messages circulent en clair entre le terminal et IMS, dans ce cas les informations peuvent être lues ou modifiées par un attaquant. Une série d'attaques peuvent exploiter cette vulnérabilité par exemple : écoute passive ou active, man-in-the-middle…

- Manipulation des données : au niveau du réseau de cœur l'interface Zb est optionnelle, il se peut qu'elle ne soit pas implémentée par l'administrateur du réseau. Dans ce cas, les clés CK & IK circulent en clair et peuvent être lues ou modifiées.
- Accès au service non autorisé : Au niveau du premier schéma si un attaquant envoie une requête d'enregistrement incorrect, S-CSCF envoie un message SAR au HSS avant de vérifier la correspondance entre l'identité de l'utilisateur (IMSI et l'IMPI). De cette manière, l'attaquant peut accéder au service IMS, même si pour peu de temps, puisque, après la réception du message SAA du HSS, S-CSCF se rend compte que cet utilisateur n'est pas autorisé à accéder aux services IMS. S-CSCF informe alors HSS avec un autre message SAR pour rétablir l'état précédent de l'utilisateur. En utilisant cette vulnérabilité un attaquant peut envoyer un message REGISTER au S-CSCF visant le de-enregistrement d'un IMPI déjà enregistré, ce qui pourrait produire une interruption de service tandis que S-CSCF vérifie ou rétablit l'état précédent.

4.3. Protocole proposé: PKI-based One-Way IMS-AKA

On suppose que l'utilisateur a réussi l'authentification au niveau GPRS (niveau paquet). Par la suite l'utilisateur peut demander l'accès aux services IMS et ainsi procéder aux procédures d'enregistrement/authentification IMS en utilisant notre protocole IMS-AKA amélioré. La conception de notre protocole est passée par deux phases en utilisant Diffie-Hellman [Resc99], et après basée sur les PKI.

4.3.1. Protocole proposé version I: Utilisation du Diffie-Hellman

On suppose que le terminal partage avec le réseau IMS les algorithmes à utiliser pour le chiffrement et le hachage, et on suppose aussi que les CSCF et HSS partagent en plus des algorithmes de chiffrement (tel que AES [Fran03] par exemple) et de hachage (tel que SHA-1 [MaGl98] par exemple) un nombre premier p et une base g.

A la différence de l'IMS-AKA, dans cette proposition [MAFA11, Moha11], la RAND est généré par l'équipement utilisateur, en tant que vecteur de RAND ordonné, le terminal extrait une valeur RANDi de ce vecteur à chaque fois qu'il veut envoyer une demande d'enregistrement. En plus, CKi et Iki sont également utilisés pour sécuriser la requête SIP REGISTER entre le terminal et S-CSCF.

Lorsque P-CSCF reçoit la demande du terminal, il commence une négociation Diffie-Hellman avec S-CSCF. P-CSCF ajoute une valeur α au message avant la transmission au S-CSCF, où: $\alpha = g^a \bmod p$ (*a* est un nombre aléatoire). En outre, cela se fait entre S-CSCF et HSS, afin d'obtenir la réponse HSS contenant les informations AV.

Cette approche résout certains problèmes de sécurité, elle assure l'authentification mutuelle entre le terminal et S-CSCF, la propriété d'accord sur les clés est également assurée. L'utilisation des RANDi entre le terminal et HSS réduit la vulnérabilité d'une attaque de rejeu.

D'autre part, l'échange de clés Diffie-Hellman est vulnérable à une attaque man-in-the-middle: cette vulnérabilité est présente parce que l'échange de clés Diffie-Hellman n'authentifie pas les participants. Cela signifie que le problème de la manipulation de données n'est toujours pas résolu.

4.3.2. La version améliorée du protocole proposé : PKI-based One-way IMS-AKA

Afin de résoudre les problèmes de sécurité de la première approche, celle-ci est basée sur la cryptographie à clés publiques, utilisée au niveau du réseau de cœur IMS. En plus des algorithmes de chiffrement (comme AES [Fran03, Jonp04]) et de hachage (SHA-1 par exemple [East06]) partagée entre le terminal et le HSS, et entre les CSCF et le HSS, nous supposons également que chacun des P-CSCF, S-CSCF et HSS dispose d'un certificat numérique. En outre, chaque élément du réseau IMS dispose d'une liste contenant son certificat numérique ainsi que les certificats des autres éléments du réseau IMS. Ceux-ci serviront aux opérations de chiffrement / déchiffrement et de signature / vérification de la signature des données transmises dans le réseau IMS.

Supposons également que le HSS joue le rôle de l'autorité de certification de confiance dans le réseau IMS, il génère donc des certificats pour le P-CSCF, le S-CSCF et pour lui-même (son certificat sera auto-signé avec sa clé privée). Le HSS doit aussi prendre en considération la validité des certificats générés, ce qui se traduit par le fait que le HSS doit générer et envoyer de nouveaux certificats au P-CSCF et au S-CSCF chaque fois que la période de validité de ces derniers expire.

Nous avons utilisé les mêmes fonctions cryptographiques spécifiées en IMS-AKA pour minimiser les changements apportés à l'architecture du système, Le protocole proposé, illustré sur la figure 4.12, est composé des étapes suivantes:

P1 (Message M1) : La première fois où l'utilisateur se connecte au réseau il génère un vecteur de nombre aléatoire RAND. Ce vecteur contient n valeur aléatoire RANDi ordonnées. A chaque demande d'authentification, l'utilisateur choisit la valeur RAND suivante pour défier le serveur HSS.

A l'étape i du protocole proposé le terminal prend un nombre aléatoire RANDi à partir du vecteur RAND.

$$RAND_i = RAND[i] \qquad (1)$$

Si le terminal n'a pas un vecteur de RAND il génère un nouveau (dans ce cas i=0 i.e. $RAND_i = RAND[0]$). A noter que l'échange du vecteur RAND se fait d'une façon sécurisée en utilisant *CKi* et *IKi* (Message M1 de la figure 4.12).

Le terminal calcule ensuite la valeur RESi et dérive les deux clé CK et IK comme suit :

$$RES_i = f_{2k(RAND_i)}, \qquad CK_{i=} f_{3k(RAND_i)}, \qquad IK_{i=} f_{4k(RAND_i)} \qquad (2)$$

Avec :

k est la clé secrète pré-partagée entre le terminal et le HSS;

fi sont les algorithmes cryptographiques partagés entre l'UE et le HSS: *f2* est une fonction de génération du code d'authentification de message, *f3* et *f4* sont des fonctions de génération de clés.

Par la suite le terminal envoi une requête SIP REGISTER contenant son IMPI, RANDi et RESi. Cette requête parvient au S-CSCF via le P-CSCF. *CKi* et *IKi* sont utilisés pour assurer la confidentialité et l'intégrité des données entre le terminal et le S-CSCF, en chiffrant les informations critiques et en authentifiant toutes les informations qui ne seront pas modifié pendant la transmission.

Par information critique nous voulons dire toute information utilisée dans l'authentification, en particulier la réponse RESi et le champ *Security-Server*. Pour la protéger, le terminal envoie un message SIP en mode de transport (même logique que IPsec en mode transport à la fois avec ESP et AH). La Figure 4.11 explique le format d'un message SIP envoyé par le terminal.



Figure 4.11. La requête SIP REGISTER envoyé par le terminal IMS

P2 (Messages M2-M5) : P-CSCF génère une clé AES, qui sera utilisée pour chiffrer les communications entre le P-CSCF et le S-CSCF concernant cet utilisateur lors de la session en cours (AES_{PS}), puis chiffre la clé AES_{PS} générée avec la clé publique du S-CSCF. Le P-CSCF ajoute la clé ensuite AES_{PS} chiffrée ainsi et sa signature et transmet le message au S-CSCF via I-CSCF.

P3 (Message M6) : S-CSCF reçoit la requête et vérifie la signature du P-CSCF. Si la signature est authentique, il déchiffre la clé AES_{PS} utilisant sa clé privée. Supposons que S-CSCF n'a pas les AVs pour cet utilisateur, alors S-CSCF ajoute sa signature au message MAR (*Multimedia Authentication Request*) et l'envoie au HSS.

P4 (Message M7) : Le HSS vérifie la signature du S-CSCF. Si elle est valide, il utilise IMPI pour trouver la clé secrète pré-partagée avec cet utilisateur (K) ainsi que le vecteur RAND, si

la valeur RANDi est différente de celle attendue, alors le HSS arrête la procédure d'authentification et informe le S-CSCF qui va envoi à l'utilisateur un message SIP 401 UNAUTHORIZED pour lui informer que l'authentification est échouée. Sinon, dans le cas contraire, *i.e.*la valeur est celle attendue, le HSS calcule l'AVi. Le HSS génère ensuite une clé AES qui sera utilisé pour sécuriser la communication entre le S-CSCF et le HSS jusqu'au prochain échange MAR/MAA (AES_{HS}). Le HSS ajoute par la suite la clé symétrique et chiffre le message de MAA en utilisant le certificat du S-CSCF. Enfin, il signe le message avant de l'envoyer au S-CSCF.

P5 (Verification 1) : Après avoir reçu le message de MAA, S-CSCF vérifie la validité de la signature du HSS. Si elle est authentique, il déchiffre le message et extrait AVi. Le S-CSCF vérifie ensuite, en utilisant IKi, la valeur de hachage reçu à l'étape P2 (calculé par le terminal) pour vérifier l'intégrité de la demande, si le résultat est positif le S-CSCF déchiffre la réponse RESi (en utilisant CKi reçu dans le vecteur d'authentification AVi). RESi est comparée à XRESi contenue aussi dans AVi. Si elles sont égales, cela signifie que cet utilisateur est légitime.

P6 (Messages M8-M10) : Après les échanges SAR/SAA avec le HSS, le S-CSCF prépare un message SIP OK. Ce message comprend AUTNi, CKi et Iki, chiffrée avec la clé AES_{PS}. Ensuite, il le signe et l'envoie au P-CSCF.

P7 (Message M11) : P-CSCF vérifie la signature du S-CSCF. Si le résultat est positif, il déchiffre le message, stocke CKi, IKi et le champ "*Security-Client*", puis transmet le message SIP OK avec AUTNi à l'utilisateur. Ce message est chiffré et authentifié par CKi et Iki (comme dans l'étape P1).

P8 (Verification 2) : Le terminal calcule AUTNi et le compare à celui obtenu à partir du S-CSCF. Le S-CSCF est alors authentifié si le résultat est positif.

Notons que nous utilisons également les deux champs *Security_Client* et *Security_Server* afin d'établir une association de sécurité IPsec entre le terminal et le P-CSCF.

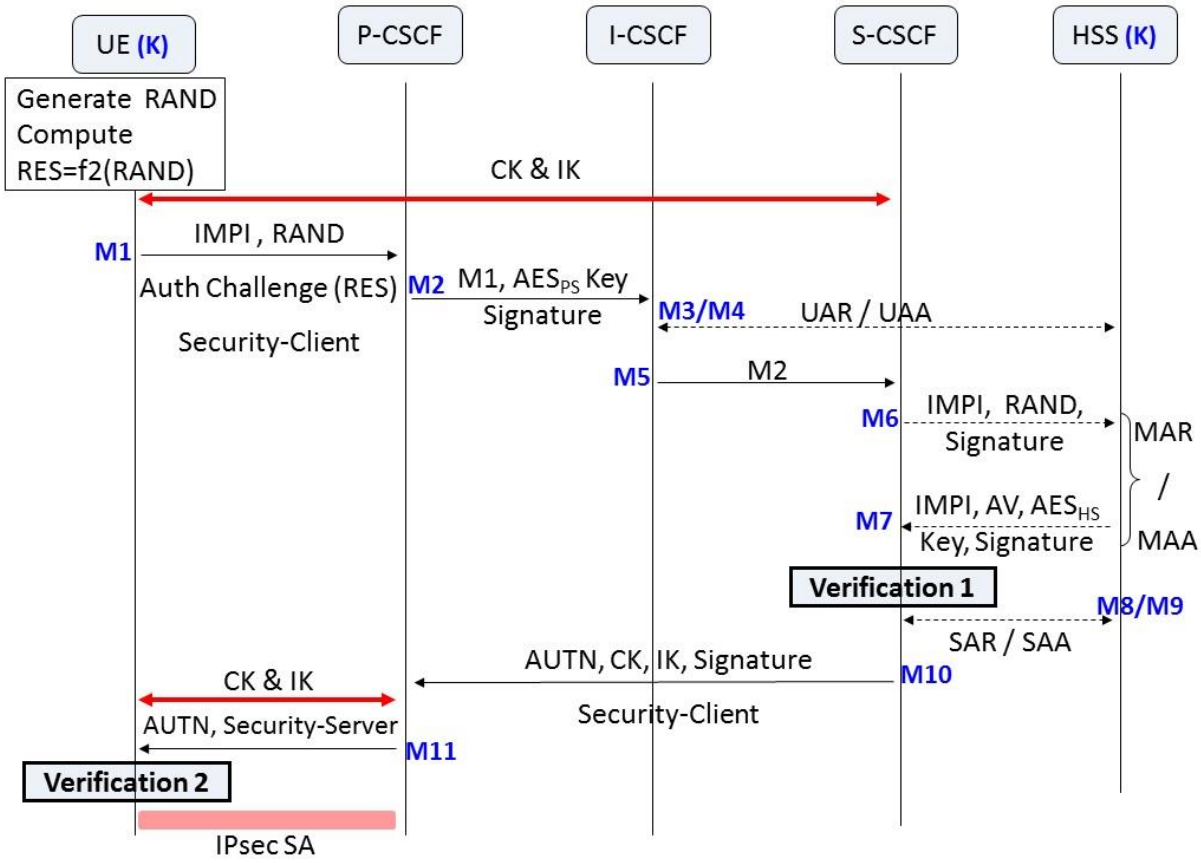


Figure 4.12. Protocole proposé : version basée sur les PKI

4.4. Implémentation

4.4.1. Outils utilisés

Le projet vise à mettre en œuvre les fonctions du CSCFs et du HSS spécifiques à notre protocole proposé. Pour ce faire, nous avons utilisé *LittleIMS*, une implémentation « Open Source » de plusieurs modules du réseau IMS/TISPAN. Pour la partie cryptographique du projet, nous avons utilisé la bibliothèque *BouncyCastle* et le paquet *java.security*. Une brève description de ces technologies sera présentée dans les sections suivantes.

4.4.1.1. Little IMS

LittleIMS [Litt00] est une implémentation open source de plusieurs éléments du réseau IMS/TISPAN. Le but de LittleIMS est de fournir la plupart des fonctionnalités présentes dans un réseau IMS, de façon simple et extensible. LittleIMS est également une application de référence montrant comment développer des applications convergentes SIP/HTTP/Diameter utilisant Cipango et la pile d'applications Spring/Hibernate/Wicket.

Cipango est une extension Servlet SIP du servlet Jetty HTTP. Cipango est donc un serveur d'applications SIP les deux standards SIP 1.1 et HTTP 2.5. Il dispose également d'une extension de diamètre pour développer des applications IMS [Cipa00].

LittleIMS est composé des éléments suivants (Figure 4.13) :

- Un S-CSCF qui offre :
    - une interface Mw
        - pour permettre aux terminaux IMS de se connecter au S-CSCF,
        - pour inter-opérer avec les CSCFs
        - une interface ISC pour permettre l'intégration de serveurs d'applications,
    - une interface pour inter-opérer avec MRFC
    - Une interface Cx Diameter pour se connecter au HSS
- Une base de données HSS qui offre:
    - Une interface Cx Diameter pour se connecter au I-CSCF et S-CSCF
    - Une interface de provisionnement Web
- Un P-CSCF
- Un I-CSCF
    - Une interface Cx Diameter pour se connecter au HSS

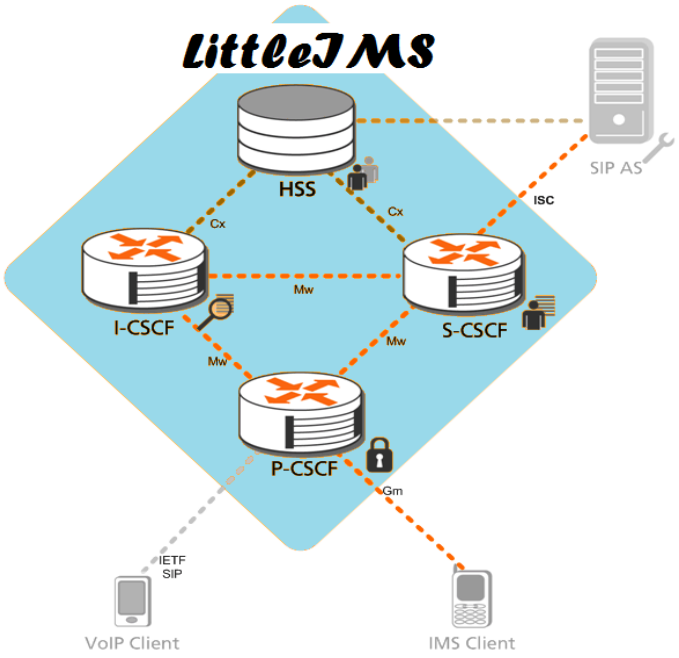


Figure 4.13. Les composants du LittleIMS

LittleIMS est un projet Maven écrit en Java, un avantage important de l'utilisation de cette application est donc sa portabilité. La configuration de chacun des éléments est réalisée par une interface web. Le tableau 4.2 présente l'URL utilisée pour configurer chaque élément du réseau de cœur IMS (le HSS et les différents CSCFs).

Tableau 4.2. Interfaces Web de configuration

Elément	Nom du Réseau	URL de Configuration web
HSS	hss.cipango.voip	http://hss.cipango.voip:8080
P-CSCF	p-cscf.cipango.voip	http://p-cscf.cipango.voip:8070
S-CSCF	s-cscf.cipango.voip	http://s-cscf.cipango.voip:8090
I-CSCF	i-cscf.cipango.voip	http://i-cscf.cipango.voip:9000

4.4.1.2. Cipango

Selon la description donnée sur *en.wikipedia.org,* Cipango est une extension VoIP du conteneur Jetty HTTP Servlets. Il est l'un des trois conteneurs Open Source SIP Servlets compatible avec la dernière servlets SIP 1.1 standard. Il offre également un soutien de protocole Diameter.

Jetty est un serveur HTTP et un moteur de servlet entièrement fondé sur la technologie Java. Jetty est un logiciel libre distribué selon les termes de la licence Apache 2.0. Il est utilisé par plusieurs autres projets populaires comme les serveurs d'applications JBoss et Geronimo.

En raison de sa petite taille, il convient parfaitement pour fournir des services web une fois embarqué dans une application Java. Depuis 2009, le développement du noyau est hébergé par la fondation Eclipse. Il est embarqué dans la distribution Eclipse en tant que greffon.

4.4.1.3. Maven

Apache Maven est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier. L'objectif recherché est comparable au système *Make* sous Unix : produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

Il est semblable à l'outil Ant, mais fournit des moyens de configuration plus simples, eux aussi basés sur le format XML. Maven est géré par l'organisation *Apache Software Foundation*. Précédemment Maven était une branche de l'organisation Jakarta Project.

Maven utilise un paradigme connu sous le nom de Project Object Model (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production. Il est livré avec un grand nombre de tâches prédéfinies, comme la compilation de code Java ou encore sa modularisation.

Un élément clé et relativement spécifique de Maven est son aptitude à fonctionner en réseau. Une des motivations historiques de cet outil est de fournir un moyen de synchroniser des projets indépendants : publication standardisée d'information, distribution automatique de modules jar. Ainsi en version de base, Maven peut dynamiquement télécharger du matériel sur des dépôts logiciels connus. Il propose ainsi la synchronisation transparente de modules nécessaires.

4.4.1.4. Outils cryptographiques

Pour la partie cryptographique du projet, nous avons utilisé la bibliothèque *BouncyCastle* [Boun00] et le paquet *java.security*.

BouncyCastle est une collection d'API utilisés en cryptographie. Il comprend à la fois des API en Java et en C#. L'architecture BouncyCastle se compose de deux éléments principaux qui implémentent les algorithmes cryptographiques de base. Ils sont connus sous les noms de l'API *light-weight*, et le fournisseur JCE.

Le paquet java.security fournit les classes et les interfaces nécessaires pour le framework de la sécurité. Cela inclut les classes qui implémentent des fonctions de contrôle d'accès. Ce paquet prend également en charge la génération et le stockage des paires de clés cryptographiques publiques, ainsi qu'un certain nombre d'opérations cryptographiques exportables y compris ceux pour le calcul des empreintes et pour la génération des signatures. Enfin, ce paquet fournit des classes qui prennent en charge les fonctions permettant la génération des nombres aléatoires.

4.4.2. Détails de l'implémentation

Pour implémenter les fonctions de CSCF et HSS requis par notre one-way AKA , nous avons modifié certains fichiers importants dans le code source LittleIMS.

Ci-dessous sont présentées quelques observations au sujet de l'implémentation :

1. Pour chaque signature ajoutée au message SIP, il a été choisi et signé uniquement la partie du message dont la valeur ne change pas au cours de la transmission (par exemple: From, To, Contact, Call-ID, tête d'autorisation, etc, mais pas: Via et Max-Forwards).
2. La clé AES générée par chaque élément de réseau pour le chiffrement et la signature a été concaténée à la valeur nonce, en étant séparées par une virgule. Nous avons choisi cette approche (et ne pas créer de champs additionnels au message) afin de ne pas modifier la structure des messages SIP.

La figure 4.14 présente le champ *Authorization*, qui contient des champs importants d'une requête SIP REGISTER envoyée de P-CSCF à S-CSCF (sa structure peut être observée à l'aide de l'interface Web du P-CSCF).

```
Authorization: Digest
    response="acde3f217597bee040157748e983135b6f1fc23a2027490
            52be76dd834db8df2",
    username="bob@cipango.voip",
    hash="18c20990a32d7b7c566cfeb372b254a7e87c9bb94a32d024cc4
            ad85f20f034c1",
    nonce="9d284362743c53ae8ae20c99b2a97a2c,402a75235df088712
            14f0d1d973d5aadb7e60f6aacb7d5b644a1ee77370c17c34b23f7b8d6
            2fef15ae6fb5dff6118e6205b258512f191c53acd0508e4c131df6fbe
            038c4bf98790fc166ec1fa0b9b4f28a582509257837d4ac3e092dc9c0
            2110aaef8895d044740f735cb2d377d4f66e3c15a337cc14e95321259
            16b5b9c9ad9,0149500316137f2796575f52ca76e42889f78d6b0e325
            97dad48d5e5c3aed064ebaf6f870e3e49453bab2bc7f4f70915ad9e59
            0738b99c0451a1f6f5c8ca5bf2f467773b74a0cc8f6ed4806b7ffed78
            f33ce4c85074c4905371061f9b3f63d89ec7f54eef99507f9a51d00be
            4f099d92b07f206494634d884e9a456d88901e77",
    realm="dfaf030a2dc839d3f50e5f5f0bd29484",
    uri="87616e8523d1ee20be08fdb608432eb314f3ead551d30a661626dc4b8
            a38b51a",
    algorithm="AKA_ONEWAY_PKI"
```

Figure 4.14. Le champ "Authorization"

Pour observer le flux des messages, nous avons utilisé Wireshark [Wire00] et les interfaces Web. Dans la capture présentée ci-après (figure 4.15), une requête SIP REGISTER envoyé par le terminal est illustrée. On peut observer quelques champs importants qui sont les suivants: l'algorithme utilisé par le client, le champ nonce qui contient la valeur Randi généré par le client et aussi la réponse calculée par le client (sur la base de la clé et de fonctions pré-partagée entre le client et HSS).

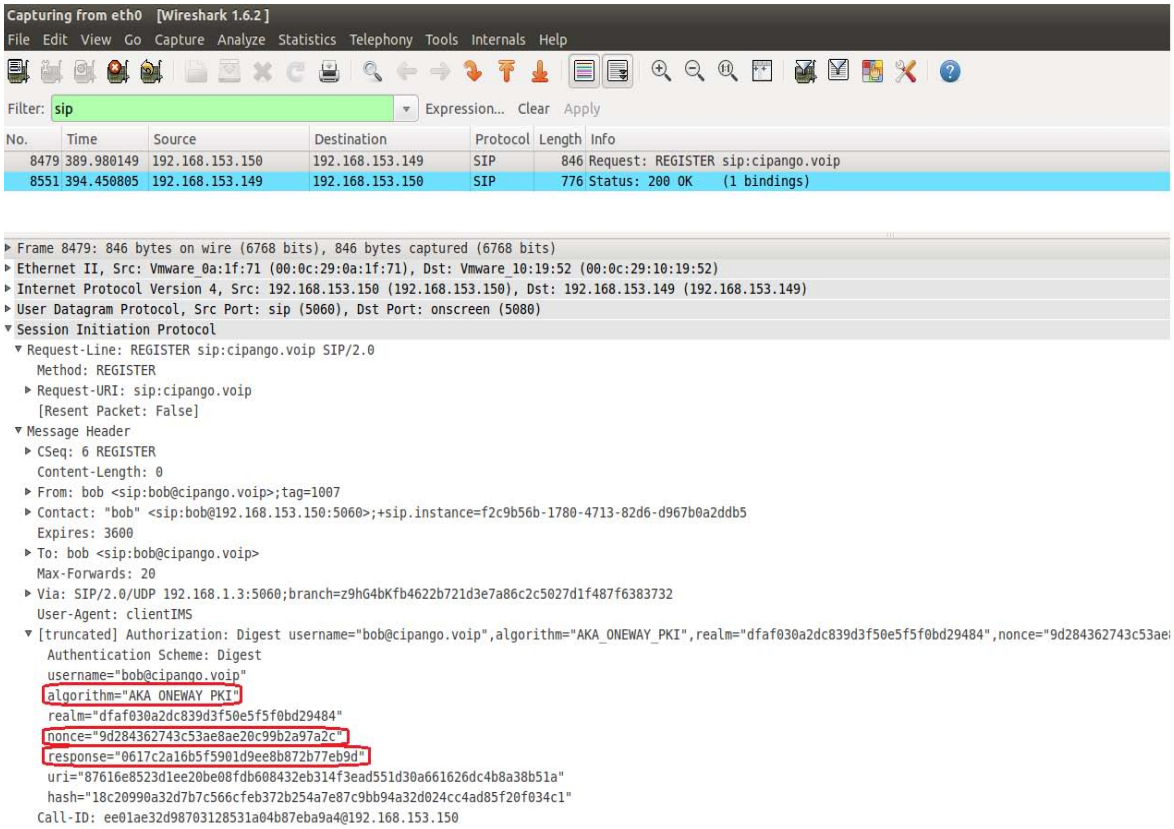


Figure 4.15.Wireshark Capture

En utilisant les interfaces web pour chaque élément, nous pouvons observer (Figure 4.16) plus de détails concernant le réseau IMS, comme la structure des messages transmis dans le réseau de base ou les identités enregistrées par HSS (Figure 4.17).

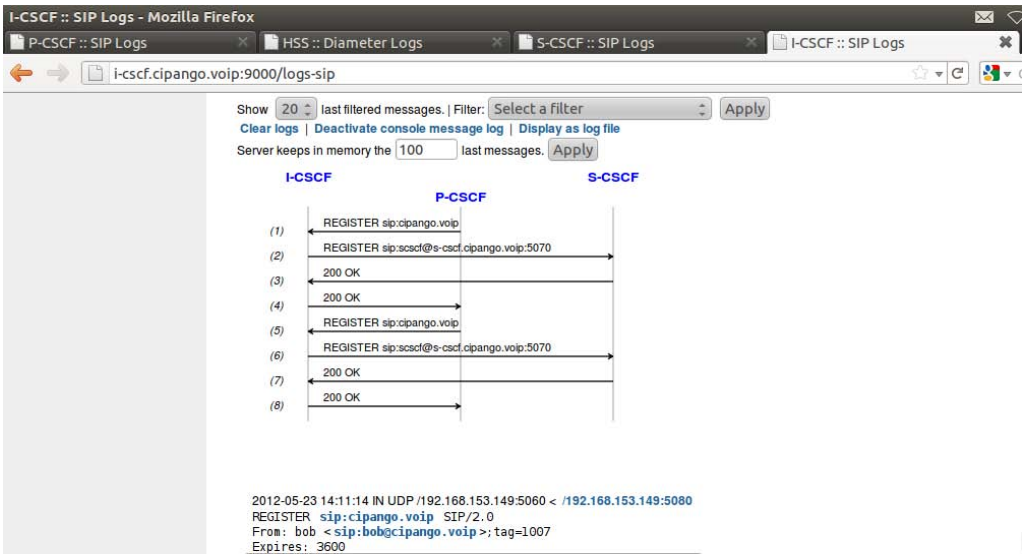


Figure 4.16. Le flux de messages SIP au niveau du I-CSCF visualisé sur l'interface web du I-CSCF

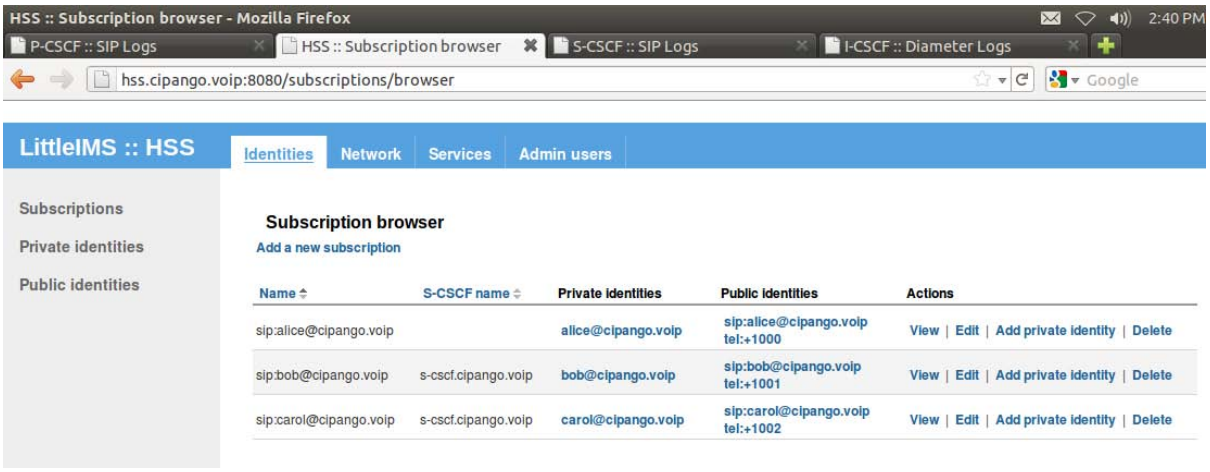


Figure 4.17. Les identités enregistrées visualisées sur l'interface web du HSS

4.5. Analyse de performances et de sécurité

Dans section nous allons évaluer les performances de notre protocole proposé sous trois angles. Premièrement, vérifions les propriétés de sécurité assurées par le protocole, cette vérification est confirmé l'outil AVISPA. Deuxièmement, nous comparons notre protocole avec le protocole de référence IMS-AKA. Cette comparaison se base sur le nombre de messages échangés lorsqu'ils sont transmis à travers une interface d'air. Finalement, la troisième méthode consiste à comparer notre protocole et IMS-AKA en termes de consommation de la bande passante.

4.5.1. Analyse de sécurité

Cette section représente une analyse du mécanisme proposé en termes de sécurité. Certaines des solutions aux problèmes de sécurité sont renforcées à partir de notre première proposition (basé sur Diffie Hellman). En ce qui suit nous présentons les propriétés de sécurité vérifiés par notre protocole :

- Authentification mutuelle entre le terminal et S-CSCF (IMS Network): du point de vue réseau IMS, S-CSCF récupère le vecteur d'authentification (AV) du HSS et vérifie que le terminal est bien authentifié en comparant XRESi avec RESi envoyé par le terminal. Du point de vue utilisateur, le terminal compare AUTNi reçu du S-CSCF avec celui qu'il calcule.
- Propriété d'accord sur les clés (Key agreement) : CK et IK sont négociés afin d'établir une session IPsec entre le terminal et P-CSCF après l'authentification. De plus, dans notre approche, ces deux clés sont utilisées lors de la procédure d'authentification/d'enregistrement entre le terminal, S-CSCF et P-CSCF (étape P1 et P7).
- Confidentialité et intégrité des données: la confidentialité et l'intégrité des données transitant entre le terminal et le réseau IMS sont assurées par le chiffrement et le hachage des messages SIP en utilisant les clés CK et IK avec des algorithmes de chiffrement/hachage pré-partagés. Grâce à notre proposition avec des techniques de clés publiques, le problème de la manipulation des données est vraiment résolu dans le réseau de cœur, puisque les informations échangées sont chiffrées et signées entre les CSCF.
- Non-répudiation dans le réseau de cœur : cette propriété est assurée puisque chacun de P-CSCF, S-CSCF et HSS possède maintenant un certificat numérique, et ajoute sa signature au message qui veut envoyer.
- Attaque de rejeu de session : maintenant c'est le terminal qui génère le RAND, un utilisateur usurpateur peut lancer une attaque de rejeu, ceci n'est pas possible. En effet, les valeurs RAND sont ordonnées entre le terminal et le HSS, en plus le HSS ne répond que si la valeur RANDi est celle attendue. D'autre part, la réponse AUTN est calculée en utilisant un numéro de séquence SQN (même calcul qu'IMS-AKA). De cette manière, le protocole peut résister aux attaques par rejeu dans les deux sens (terminal et serveur HSS).

Nous avons validé le protocole proposé (*The one-way IMS-AKA*) en utilisant AVISPA [Avis00]. AVISPA est un outil *push-button* avec des technologies industrielles très puissantes pour l'analyse des différents protocoles et applications de sécurité Internet. Il est utilisé par les

développeurs de différents protocoles de sécurité et aussi par des chercheurs universitaires. Nous avons construit un modèle AVISPA pour la séquence des messages du protocole *one-way* IMS-AKA.

```
Using Ks and a random number (nonce) RAND U generates:

      RES = F2(Ks.RAND)    where F2 hash
      CK  = F3(Ks.RAND)    where F3 one-way
      IK  = F4(Ks.RAND)    where F4 one-way
U -> S: {U.RAND.{RES}_CK}_IK

     Same, using the received RAND, S generates

      XRES =  F2(Ks.RAND)    where F2 hash     % (XRES: eXpected RES)
      CK   =  F3(Ks.RAND)    where F3 one-way
      IK   =  F4(Ks.RAND)    where F4 one-way
      AK   =  F5(Ks.RAND)    where F5 one-way
      AUTN =  {seq}_AK.F1(Ks.seq.RAND)  where F1 hash


      S compares RES with XRES, if OK, the user U is authenticated,
      then S responds:

S -> U: {{AUTN}_CK}_IK

     U  checks  AUTN,  if  OK,  the  server  is  authenticated,  and  U
increments his Seq number.
```

Figure 4.18. Le modèle AVISPA utilisé pour vérifier le protocole PKI-based one-way IMS-AKA

La figure 4.18 montre les champs du message modélisés par le modèle AVISPA utilisant la notation traditionnelle d'Alice et Bob. Deux entités concernées, le terminal et S-CSCF, sont représentés par U et S. S et U partagent une clé secrète Ks et chacun d'entre eux possède une séquence (*seq*), qu'ils essayent de la maintenir synchronisée.
Le modèle AVISPA est construit pour valider deux propriétés de sécurité: le secret de la valeur de la séquence *seq*, le secret de CK et IK qui sont répartis entre le terminal (U) et S-CSCF (S), et l'authentification mutuelle entre U et S.

Après vérification en utilisant le modèle présentée en figure 4.18, le OFMC, ATSE, SATMC et TA4SP rapportent que notre modèle est bien sécurisé (SAFE), qui pourrait être interprété comme la preuve qu'aucune trace d'attaque, qui viole les propriétés de sécurité spécifiées, n'a été trouvée.

Modèle d'attaquant considéré

Une approche intuitive pour découvrir des failles dans un protocole de sécurité consiste à modéliser un intrus spécifique et à observer le comportement du protocole agissant en concurrence avec celui-ci. Par la suite, il suffit de chercher une attaque menée à terme parmi tous les dénouements possibles, et répéter au besoin cette procédure avec d'autres processus ennemis.

De toute évidence, cette approche ne convient généralement pas à démontrer qu'un protocole n'admet aucune faille. La complétude d'une telle méthode de validation requiert donc l'interaction du protocole avec un plus puissant attaquant, c'est-à-dire un processus ennemi capable de reproduire toute attaque réalisable par un autre processus ennemi. Malgré le fait que les techniques de modélisation et les algorithmes de vérification diffèrent énormément d'un auteur à l'autre, ce concept presque utopique de plus puissant attaquant est souvent approché par un seul intrus ayant la capacité de communiquer et d'interagir avec le protocole selon les règles suivantes.

- Un intrus peut intercepter tout message envoyé sur un canal public et, par conséquent, acquérir de nouvelles données qui lui permettront de construire de nouveaux messages.
- Un intrus est un participant légitime, et a donc la capacité d'initier le protocole avec tout autre participant ou, réciproquement, de recevoir des invitations de la part des autres participants.
- Un intrus possède certaines connaissances initiales incluant ses clés privées, toute clé publique accessible aux autres participants, et son propre ensemble de nonces et de messages forgés.

Le modèle que nous avons utilisé pour valider le protocole proposé est le modèle de Dolev-Yao [DoYa81, Herz05]. Dans ce modèle, un intrus peut utiliser n'importe quel terme qu'il a observé sur le réseau précédemment et ce, autant de fois qu'il le désire. On dit habituellement que "l'intrus est le réseau" ainsi, les messages, qui circulent sur le réseau, sont connus par l'intrus et constituent l'ensemble de sa connaissance initiale. Donc, dans ce système, on considère que l'intrus a le contrôle total du réseau. Il connait toutes les données publiques des agents, dispose des privilèges et des clés des agents malhonnêtes. Ainsi, l'intrus peut appliquer une des règles données par le modèle : Il peut déchiffrer un message s'il connaît la clé de déchiffrement, il peut chiffrer un message avec n'importe quelle clé en sa possession et il est capable de mémoriser, effacer, construire et envoyer tous les messages s'il a la clé.

4.5.2. Analyse de coût du nombre de messages échangés

Dans cette section, nous allons comparer en termes de performance notre protocole Version II (*PKI-based*) avec le protocole IMS-AKA normal. La comparaison est faite sur la base du nombre de messages échangés puisqu'ils sont transmis à travers une interface air. Nous adoptons l'hypothèse décrite dans [LCHW05]. L'hypothèse se présente comme suit :

Supposant que le coût d'un message entre le terminal et S-CSCF est égale à l'unité =1, et les messages sur l'interface Cx = α, dès maintenant on peut affirmer que avec $\alpha < 1$ pour les raisons suivantes :

- I/S-CSCF et HSS échangent les messages via Cx à travers un réseau IP filaire, par contre les messages SIP traversent une interface air GPRS/UMTS par exemple.
- En général les I/S-CSCF et HSS se trouvent dans la même localisation, par contre les terminaux, généralement, se trouvent dans un réseau visité (accès distant).

Soit C : le coût total du IMS-AKA, et C_{PKI} : le coût total du protocole proposé. Si S-CSCF n'a pas d'AV valide, le coût d'IMS-AKA est exprimé par C1. Sinon, si les messages MAR/MAA ne sont pas exécutées dans, le coût d'IMS-AKA est exprimée par C2:

$$C1 = 4 + 8\alpha \qquad C2 = 4 + 4\alpha \qquad (3)$$

L'enregistrement IMS est effectué d'une façon périodique. Si on suppose qu'une exécution des messages MAA/MAR permet de récupérer un nombre d'Av égale à ***n***, et si on suppose que le nombre d'opérations (authentification/enregistrement) que le S-CSCF traite est égale à ***m***. Par conséquent, seulement $\left\lceil \frac{m}{n} \right\rceil$ messages MAA/MAR sont exécutés.

Avec $\lceil \cdot \rceil$ est la partie entière supérieure. On pose $x = \left\lceil \frac{m}{n} \right\rceil$, de l'équation (3) on peut déduire que le coût moyen d'IMS-AKA C peut être exprimée comme suit:

$$C = \frac{x}{m} C1 + \frac{(m-x)}{m} C2 = 4 + \left(\frac{2x}{m} + 4\right)\alpha \qquad (4)$$

Avec le même raisonnement, nous allons calculer le coût de notre protocole proposé (PKI-based One-way IMS-AKA). Nous prenons le ***pire cas*** (n = 1), ce qui signifie que pour chaque requête SIP REGISTER S-CSCF demande un AV de HSS. De cette manière le terminal doit

envoyer un seul RAND pas un tableau et donc il y aura toujours un échange MAR/MAA entre S-CSCF et HSS.

Dans ce cas le coût du protocole proposé exprimé par C_{PKI} est :

$$C_{PKI} = 2 + 6\alpha \qquad (5)$$

A partir des équations (4) et (5) l'amélioration I du protocole proposé par rapport au IMS-AKA est donnée par :

$$I = \frac{(C - C_{PKI})}{C} = \frac{(m + x\alpha)}{2m + \alpha(x + 3m)} \qquad (6)$$

Pour la comparaison, en plus de considérer le pire cas pour notre protocole, nous allons considérer le cas optimal pour IMS-AKA :

$$m = n \qquad i.e. \qquad x = 1 \qquad (7)$$

Cela signifie que la taille de AV est égal au nombre de requêtes traitées par le S-CSCF. Il y aura donc un seul échange MAR / MAA. D'autre part, selon [NtXS10], la valeur optimale de m est 10. Donc, si nous prenons x = 1 et m = 10, l'amélioration devient I_W (*Worst case*) :

$$I_w = \frac{C - C_{PKI}}{C} = \frac{10 + \alpha}{20 + 31\alpha} \qquad (8)$$

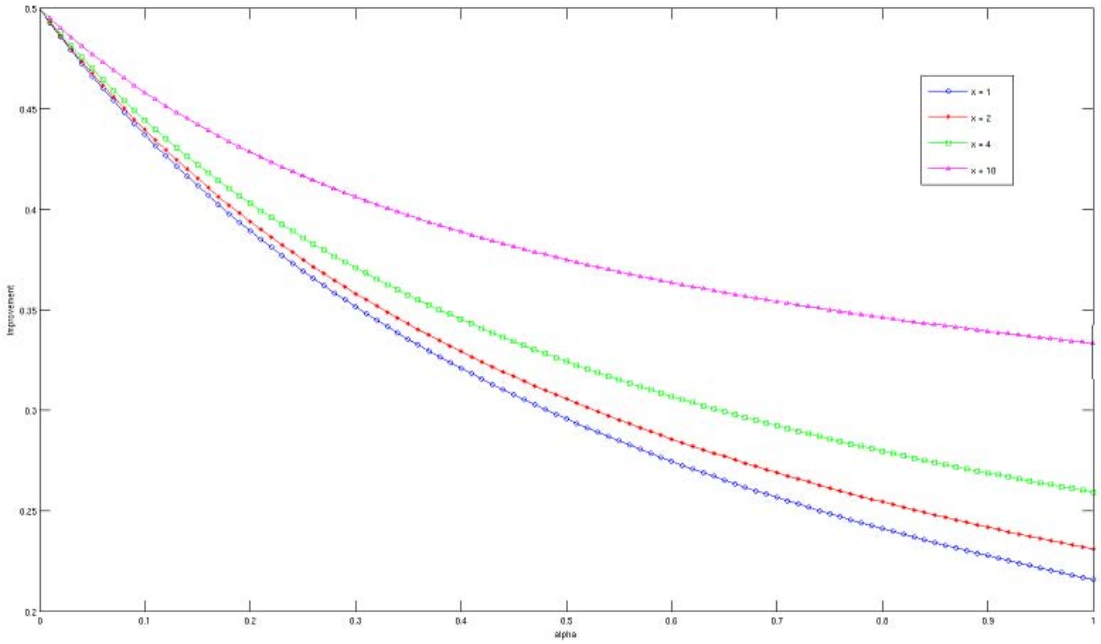


Figure 4.19. Amélioration du protocole proposé par rapport à IMS-AKA

Comme la figure 4.19 illustre, les versions proposées d'IMS-AKA peut économiser jusqu'à 50% du trafic SIP/Cx par rapport à l'IMS-AKA, et même dans le pire cas x = 1, on voit qu'avec notre protocole, nous gagnons plus de 20% (21,5%) par rapport à l'IMS-AKA.

4.5.3. Analyse de la consommation de la bande passante :

En général, toute mesure de sécurité supplémentaire introduit au moins un certain degré de surcharge protocolaire (*overhead*). Cependant, une conception judicieuse vise toujours à maintenir le niveau de la surcharge protocolaire le plus faible possible. Dans cette section, nous allons analyser la bande passante moyenne consommée par tous les messages envoyés dans le flux de messages discuté, concernant les deux protocoles : IMS-AKA et One-way IMS-AKA.

Notations : Soit Mi le message avec le numéro i envoyé dans le protocole IMS-AKA et M_{PKIi}: le $i^{ème}$ message envoyé dans la version II du protocole proposé. Observation : Pour faire cette analyse, nous n'allons prendre en compte que les paramètres de sécurité spécifiques à l'IMS-AKA (IMPI, AV, etc, mais pas: *To, From, Via, Contact*, etc.) Aussi, nous supposons, que la longueur des messages ne contenant pas des paramètres spécifiques à l'IMS-AKA est égale à q.

4.5.3.1. Analyse de la bande passante pour IMS-AKA

Les tailles des messages M1 à M20 (figure 4.5) sont calculées comme suit:

- La longueur du premier message, qui est désignée par | M1 |, est donnée par la taille de l'IMPI. Ainsi :
  |M1| = |IMPI| = 128 bits.
- Etant donné que M2 et M5 sont l'acheminement du message M1, leur longueur est la même que M1. Nous considérons que M6 a aussi la même taille, (sur la base de l'observation ci-dessus).
- La longueur de M3, M4, M13, M14, M16 - M20, est supposée être égale à q.
- M7 contient une séquence d'AV, qui est composé de RAND, XRES, CK, IK, et AUTN. Sa longueur peut être représentée comme suit:
  |M7| = | AV | = | RAND | + | XRES | + | CK | + | IK | + | AUTN | = 608 bits
  Nous supposons que le message MAR transport un seul AV.

- M8 se compose de RAND, IK, CK et AUTN = (SQN ⊕ AK_AMF_MAC). Sa longueur peut être obtenue en additionnant | RAND | + | IK | + | CK | + | AUTN | (544 bits). On peut observer que les longueurs des M9 et est la même que la longueur du M8.
- | M10 | = | RAND | + | AUTN | = 288 bits
- M11, M12 et M15 ne contiennent que 64 bits (longueur de la réponse RES).

La bande passante totale consommée peut être calculée comme suit:

$$Bw = \sum_{i=1}^{20} |M_i| = 4*128 + 9q + 608 + 2*544 + 288 + 3*64 = 2688 + 9q \text{ bits}$$

4.5.3.2. Analyse de la bande passante pour le protocole proposé

La longueur des messages du protocole One-way IMS-AKA basé sur les PKI (illustrés sur la figure 4.12) sont calculées comme suit:

- $M_{PKI\ 1}$ est composé des paramètres IMPI, RANDi, RESi et la signature du terminal en utilisant la clé IK.

$$|M_{PKI\ 1}| = 128 + 128 + 64 + 160 = 480 \text{ bits.}$$

- $M_{PKI\ 2}$ transport en plus du message $M_{PKI\ 1}$, la signature P-CSCF et la clé AES_{PS} générée par P-CSCF.

$$|M_{PKI\ 2}| = 480 + 128 + 160 = 768 \text{ bits}$$

- La longueur des messages M_3, M_4, M_8 and M_9 est supposée être égale à q.
- $M_{PKI\ 5}$ ne fait que transmettre le message $M_{PKI\ 2}$, donc sa taille est la même que celle du message $M_{PKI\ 2}$
  $|M_{PKI\ 5}| = |M_{PKI\ 2}|$.
- $M_{PKI\ 6}$ contient IMPI, RAND et la signature du S-CSCF.

$$|M_{PKI\ 6}| = 128 + 128 + 160 = 416 \text{ bits}$$

- $M_{PKI\ 7}$ contient un AV (RAND, XRES, CK, IK, AUTN) et aussi IMPI, la signature du HSSet la clé AES_{HS} générée par HSS.
- $|M_{PKI\ 7}| = |RAND| + |XRES| + |CK| + |IK| + |AUTN| + |IMPI| + |AESkey| + |Signature| = 1024$ bits

- $|M_{PKI\ 10}|$ =|RAND| + |AUTN | + |CK| + |IK| + |Signature|= 704 bits
- $|M_{PKI\ 11}|$ =|RAND| + |AUTN | + |Signature| = 448 bits

La bande passante totale consommée pour cette approche peut être calculée comme suit:

$$Bw_{PKI} = \sum_{i=1}^{11} |M_{PKI\ i}| = 480 + 2*768 + 4q + 416 + 1024 + 704 + 448 = 4608 + 4q \text{ bits}$$

4.5.3.3. Comparaison

Pour faire une comparaison entre IMS-AKA et le one-way IMS-AKA proposé, nous calculons les valeurs de q pour lesquelles $Bw > Bw_{PKI}$:

$$Bw > Bw_{PKI} \quad ==> \quad 2688 + 9q > 4608 + 4q \quad ==> \quad q > 384\ bits$$

D'après le résultat obtenu, nous pouvons observer que si$q > 384\ bits$, alors$Bw > Bw_{PKI}$. Si on tient en compte le fait que les messages que nous avons considéré leurs longueur égale à q contiennent au moins les domaines suivants: To, From, Contact, Expire, Max-Forwards, Via, CSeq ... etc on peut en déduire que la longueur de q est d'au moins 1024 bits. Dans ce cas ($q = 1024$), nous avons :

$$\frac{Bw_{PKI}}{Bw} = \frac{8704}{11904} = 0{,}73$$

Ce qui signifie que le *one-way* IMS-AKA basé sur les PKI permet de réduire la consommation de la bande passante de 27% $(1 - 0{,}73)$ par rapport à l'IMS-AKA.

4.6. Tests d'évaluations des performances

Dans cette section, pour analyser les performances du PKI-based One-way IMS-AKA, nous avons développé deux scénarios. Dans le premier scénario, on mesure le temps de traitement dans le réseau de cœur IMS. Dans le deuxième scénario, nous mesurons le délai de bout en bout pour une requête d'enregistrement (SIP REGISTER) réussie en utilisant deux réseaux d'accès : filaires et sans fil. Dans chaque scénario, nous avons mené des expériences pour les trois protocoles: IMS-AKA normal, One-way IMS-AKA basé sur Diffie-Hellman (version I) et le One-way IMS-AKA basé sur les PKI (version II).

La figure 4.20 montre l'architecture utilisée pour les tests. Le noyau IMS, qui se compose des CSCFs et de HSS sont implémentés sur une seule machine (Machine cœur IMS). Nous avons utilisé LittleIMS (comme expliqué précédemment). Du côté de l'utilisateur, nous avons développé en JAVA, deux nouveaux clients pour les protocoles proposés (à base de DH et de PKI). Les trois clients sont installés sur trois machines différentes de même configuration (Machine client IMS).

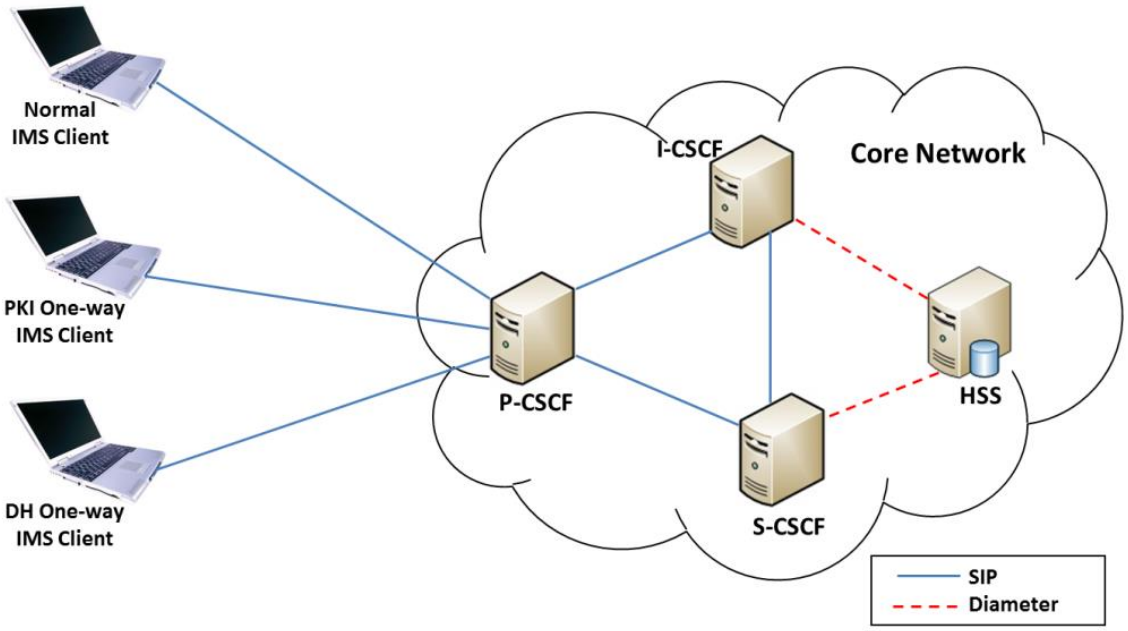


Figure 4.20. Architecture utilisée pour les tests

Les configurations des machines "cœur IMS" et "client IMS" sont données dans les deux tableaux 4.3 et 4.4 respectivement.

Tableau 4.3. Caractéristiques matérielles et logicielles de la machine "cœur IMS"

Cœur IMS	
Système d'exploitation	Ubuntu 11.10
Processeur	Intel® Pentium®
RAM	1 Go
NIC (filaire)	Ethernet 10/100 BASE-T intégré
NIC (sans fil)	ALFA awus036h

Tableau 4.4. Caractéristiques matérielles et logicielles de la machine "client IMS"

Client IMS	
Système d'exploitation	Ubuntu 12.04 LTS
Processeur	Intel Core i3-380M (2.53GHz, 3 Mo)
RAM	512 Mo
NIC1 (filaire)	Realtek RTL8167
NIC2 (sans fil)	Broadcom BCM43XX

4.6.1. Scenario 1: Temps de traitement dans le réseau de cœur IMS

Cette analyse est basée sur le temps total nécessaire pour le traitement des messages et leur propagation à l'intérieur du réseau de cœur IMS. Pour cela, nous avons calculé le temps entre les deux moments où P-CSCF reçoit le message à partir de l'équipement de l'utilisateur jusqu'à l'instant où P-CSCF renvoie la réponse vers le terminal, et on stocke la valeur de ce délai.

Pour comparer les trois protocoles, nous avons effectué une série de tests (20 tests). Chaque client des trois protocoles envoi une requête SIP REGISTER et nous avons sauvegardé les résultats concernant chaque protocole. La figure 4.21 nous montre les résultats obtenus. A noter que nous avons pris en considération seulement les requêtes d'enregistrement réussies.

Comme le montre la figure 4.21, on peut remarquer qu'il existe une différence significative entre le temps de traitement pour IMS-AKA normal et le "one-wayIMS-AKA" (les deux versions). Cela est normal puisque le protocole normal a plus de messages à traiter, et donc cela nécessite plus de temps.

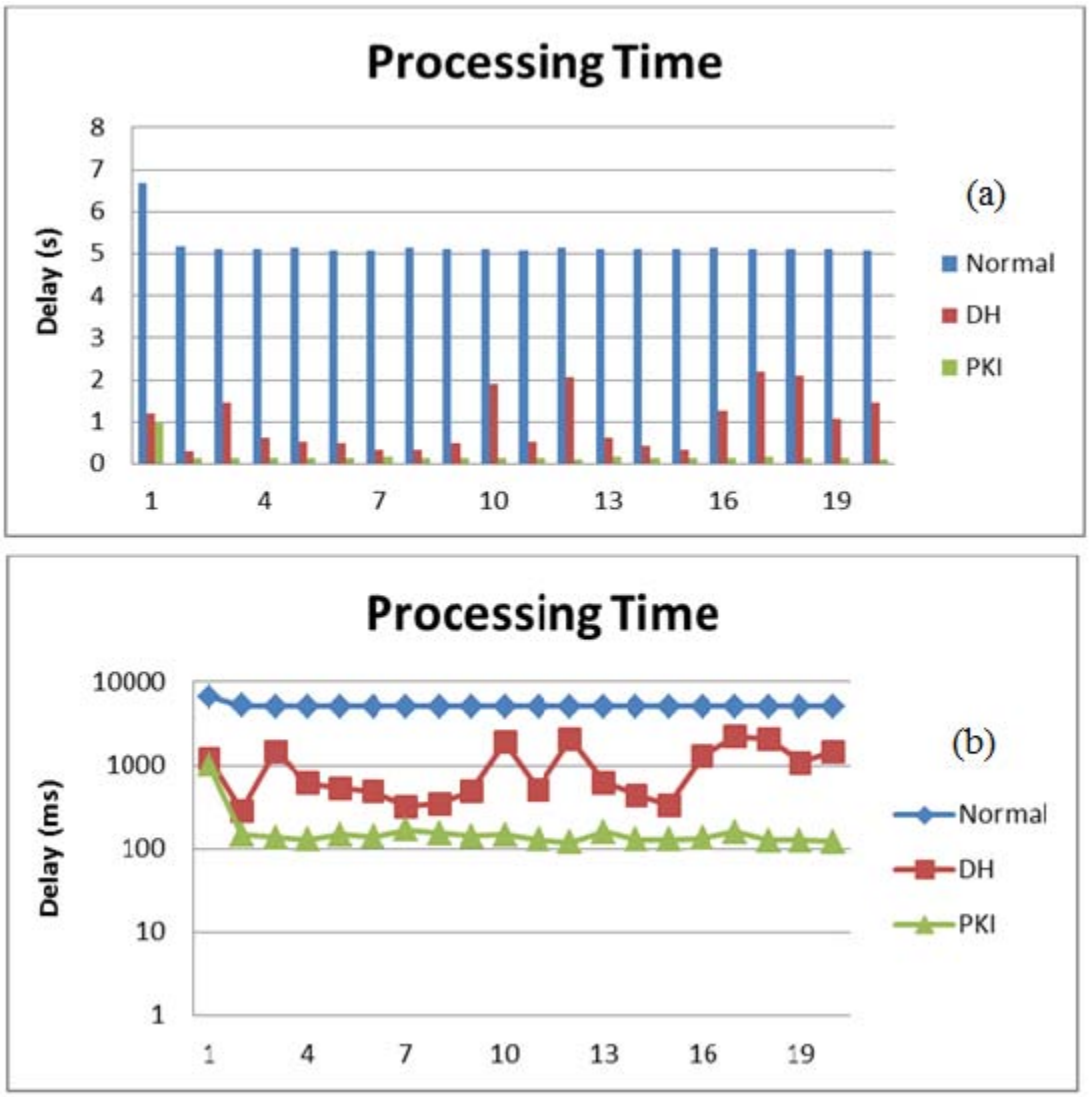


Figure 4.21. Temps de traitement au niveau du réseau de cœur

(a) échelle linéaire, (b) échelle logarithmique

Entre les deux versions proposées, le protocole basé sur les PKI est plus efficace. En fait, le protocole basé sur Diffie-Hellman prend beaucoup de temps pour générer et échanger les deux parties de la clé à utiliser plus tard pour sécuriser la communication entre P-CSCF et S-CSCF, et entre S-CSCF et HSS. Tandis que pour le protocole avec les PKI, les certificats des différents CSCF sont échangés au début de processus (essai 1), ce qui explique le pic qu'on remarque au niveau du premier essai. Par la suite les différents certificats sont stockés localement ce qui explique la faible valeur du temps de traitement puisqu'on n'a pas besoin d'attendre de les recevoir et on procède directement à la sécurisation des messages.

Nous avons calculé le temps moyen de traitement pour chaque protocole, les résultats sont présentés dans le tableau 4.5.

Tableau 4.5. Temps moyen du traitement dans le réseau de cœur

Protocole	Temps de traitement (Moyenne *ms*)
IMS-AKA	5195,05
DH	985,9
PKI	182,55

4.6.2. Scenario 2: comparaison du délai entre deux réseaux d'accès : filaire et sans-fil

L'objectif de ce scenario est de montrer l'effet d'utilisation de différents réseaux d'accès pour se connecter au réseau IMS. Nous mesurons le temps d'une requête d'authentification/registration réussie : le temps entre l'envoi du message SIP REGISTER jusqu'à la réception du message SIP OK (coté client). Nous avons effectué une serie de tests, qui consistent à envoyer des demandes d'authentification (message SIP REGISTER) en utilisant deux réseaux d'accès différents: Ethernet et WIFI. Les résultats obtenus sont présentés sur les figures 4.22 et 4.23.

A partir des résultats obtenus (figures 4.22 et 4.23) nous pouvons conclure que pour les deux réseaux d'accès le protocole IMS-AKA prend beaucoup de temps pour traiter une requête d'authentification, que les protocoles One-way (les deux versions I et II).

Dans le cas d'utilisation du WIFI (sans fil) le délai de bout en bout est amplifié, ce qui est normal puisqu'il faut attendre deux allers-retours pour recevoir la réponse.

Pour les deux protocoles one-way, PKI est plus efficace que DH. Cela est dû au temps de traitement au niveau du réseau du cœur, puisque les deux sont des one-way.

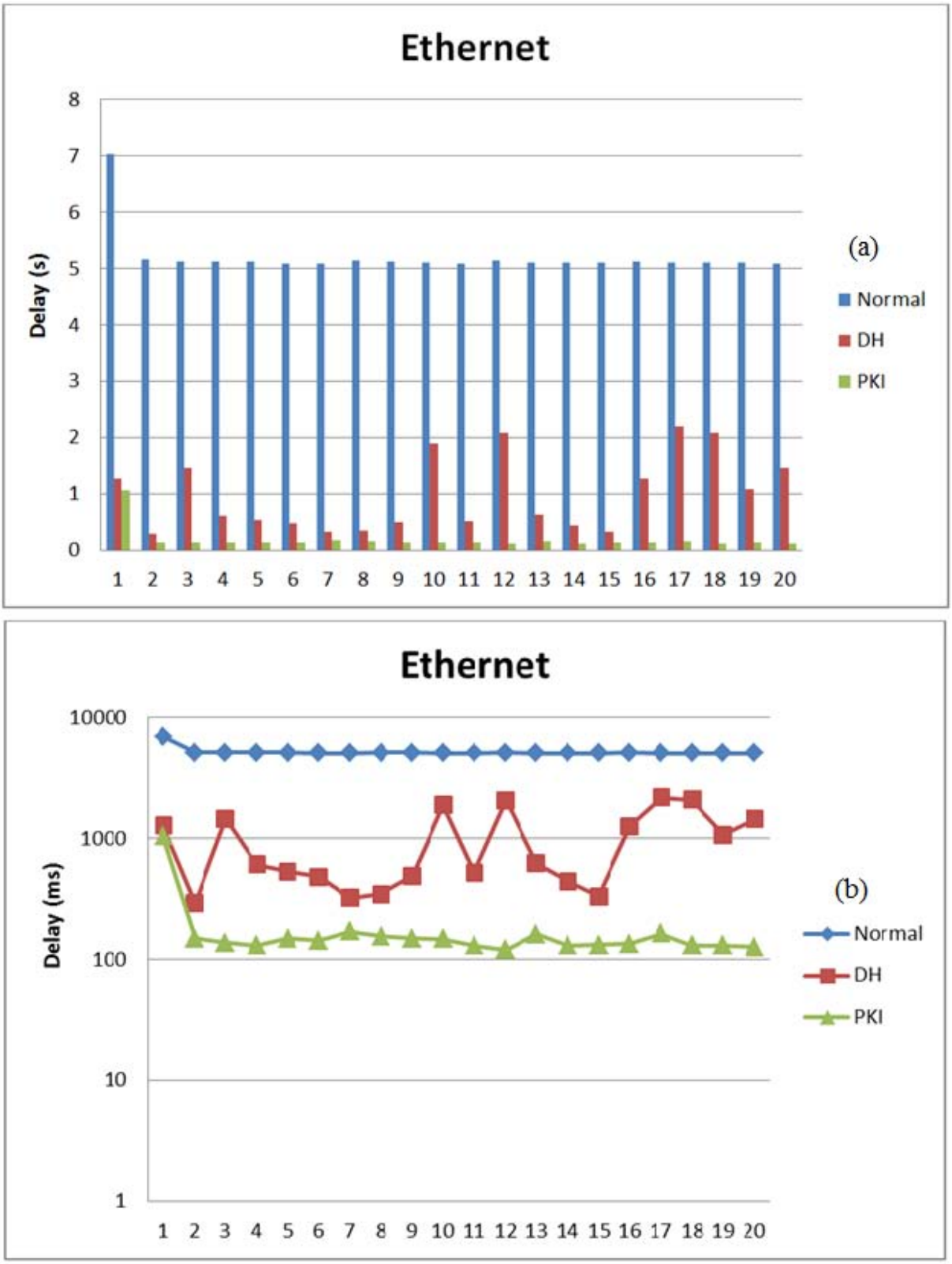


Figure 4.22. Délai d'authentification pour un réseau d'accès filaire (Ethernet)

(a) échelle linéaire, (b) échelle logarithmique

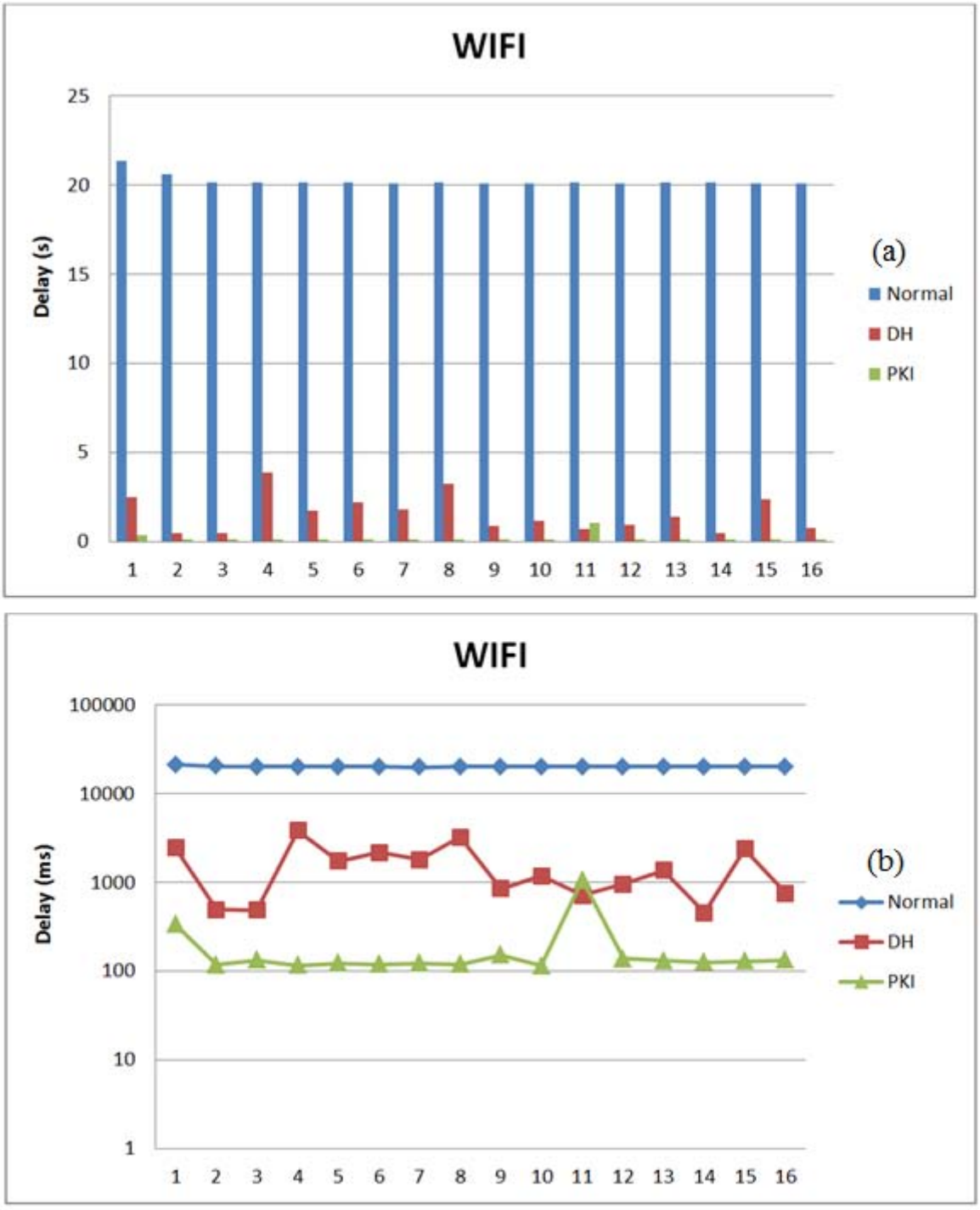


Figure 4.23. Délai d'authentification pour un réseau d'accès sans-fil

(a) échelle linéaire, (b) échelle logarithmique

4.7. Mécanismes de vérification et de révocation des certificats

Afin de gérer d'éventuelles compromissions de clés privées ou le changement d'informations du certificat et afin d'invalider un certificat avant même que sa date d'expiration ne soit atteinte, l'utilisation de méthodes de révocation sera un gage supplémentaire de sécurité permettant d'assurer la validité des statuts des certificats utilisés. La méthode de révocation permettra alors de rendre public l'état d'un certificat pour que des utilisateurs en prennent connaissance, et par conséquent refusent tout message signé ou chiffré avec la clé privée révoquée associée au certificat révoqué.

Cette solution est nettement bénéfique en termes de flexibilité et d'extensibilité i.e. adaptabilité d'un schéma de révocation appliqué au contexte d'un opérateur IMS, en mesurant le rapport de l'accroissement des coûts en mise à jour, et en requête sur l'accroissement de la taille de la communauté desservie mesurée en nombre de certificats.

Plusieurs méthodes de révocation ont été proposées pour répondre à la diversité en besoin de sécurité et en capacité de ressources (calcul, stockage et bande passante) des applications et des utilisateurs :

4.7.1. OCSP (*Online Certificate Status Protocol*)

Cette méthode, standard de l'IETF (RFC 2560) introduit un serveur OCSP de confiance. En s'adressant à ce serveur par envoi d'une requête, les vérificateurs peuvent connaitre l'état de validité d'un certificat. Le vérificateur qui a connaissance du certificat du serveur doit vérifier l'authenticité des messages signés et retournés par le serveur. Les serveurs OCSP peuvent être dissociés des ACs, voire co-localisés avec l'AC.

4.7.2. 2-3 CRT (*Certificate Revocation Tree*)

La méthode 2-3 CRT utilise un arbre d'ordre 2-3 dans lequel les feuilles correspondent aux certificats révoqués et sont ordonnées dans l'ordre croissant des numéros de série. La valeur d'un nœud de l'arbre est le hachage des valeurs de ses fils et la valeur racine est signée par l'AC pour garantir son authenticité.

Un certificat est considéré révoqué s'il apparaît comme une feuille de l'arbre; un certificat sera considéré valide si deux certificats correspondent à deux feuilles adjacentes de l'arbre avec

pour l'un, un numéro de série supérieur au certificat recherché et pour l'autre un numéro inférieur.

4.7.3. Les CRLs

L'autorité de certification AC créé une liste (de numéros de séries) des certificats révoqués nommée CRL, la date et la signe. L'AC créé une nouvelle CRL à chaque fois qu'un certificat est révoqué ou lorsque la période de validité est dépassée. Pour vérifier la validité d'un certificat, le vérificateur doit envoyer une requête au serveur de publication hébergeant la CRL correspondante, avec comme argument l'identifiant de l'AC en charge du certificat ; il reçoit alors la dernière CRL générée par l'AC ; il doit ensuite vérifier la signature de la CRL et sa durée de validité, et puis rechercher le certificat dans la CRL.

La méthode CRL a plusieurs dérivées parmi lesquelles :

4.7.3.1. Delta-CRL :

Liste signée qui contient tous les certificats révoqués depuis la publication de la dernière CRL. Ainsi, la vérification d'un certificat va nécessiter de récupérer à la fois la CRL de base et la Delta-CRL la plus récente. La Delta-CRL améliore la fraîcheur des informations de révocation, ce qui la rend plus extensible. Cependant, les coûts induits sont augmentés.

4.7.3.2. CRL Distribution Points :

Cette méthode consiste à diviser une CRL en segments contenant chacun un sous-ensemble des certificats révoqués par une AC. Comme tout certificat contient un pointeur vers le segment lui correspondant, un vérificateur va pouvoir accéder directement au segment concerné. Ainsi la méthode CRL DP apparaît plus extensible que la CRL bien qu'elle présente elle aussi un faible risque système.

4.7.3.3. Over-issued CRL :

Cette méthode réduit considérablement l'explosion des requêtes CRLs, en permettant de délivrer des CRLs de même durée de vie qui se chevauchent dans le temps. De cette façon, les CRLs n'expirent pas en même temps au niveau des vérificateurs. Cette méthode augmente le coût en mise à jour, car les CRLs délivrées sont plus nombreuses.

4.7.4. Comparaison des mécanismes de vérification de révocation de certificats

Afin de comparer ces méthodes de révocation, plusieurs critères d'évaluation [Aarn00, AJKL00, Zhen03] doivent être pris en considération :

Coût en mise à jour (CMA) : coût nécessaire à la mise à jour des informations de révocations. Ce coût se présente sous la forme d'un triplet (bw-u, OvhAC-u, OvhEAGC-u) présenté dans le Tableau 1.

Coût en requête de vérification (CR) : coût quotidien nécessaire à la validation d'un certificat. Le coût est mesuré en fonction d'un quadruplet (bw-q, OvhAC-q, OvhEAGC-q, Ovhverifier).

Tableau 4.6. Critères de comparaison

Coût en mise à jour d'un certificat	Bw-u : bande passante requise entre ACs et EAGCs OvhAC-u : coût de calcul au niveau de l'AC OvhEAGC-u : coût de calcul au niveau de l'EAGC
Coût en requête de vérification de certificat	bw-q : bande passante requise entre EAGCs et vérificateurs OvhAC-q : coût de calcul au niveau de l'AC OvhEAGC-q : coût de calcul au niveau d'une EAGC Ovhverifier : coût de calcul au niveau des vérificateurs

A ces deux critères s'ajoutent d'autres comme la fraîcheur des informations de révocation, la fenêtre de vulnérabilité (WoV : Window of Vulnerability), l'extensibilité (scalability), le risque système, déni de Service (DoS) et le schéma On-line/Off-line [Cbek04].

Dans la suite de notre étude, nous allons effectuer une comparaison entre deux méthodes : l'OCSP et les CRLs « classiques » afin d'en déduire la plus adaptée au contexte d'IMS.

Les différents coûts peuvent être modélisés par des équations (nous utilisons les mêmes notations que [Cbek04] :

Tableau 4.7. Définition des coûts

	CRL	OCSP
$bw-u$	$T.(n.p.\frac{t}{2}.l_{sn}+\frac{n}{k}.l_{sig})$	0
Ovh_{AC-u}	$T.\ c_{sig}$	0
Ovh_{EAGC-u}	$T.\ c_{verify}$	0
$bw-q$	$q.(k.p.\frac{t}{2}.l_{sn}+l_{sig})$	$q.\ l_{sig}$
Ovh_{AC-q}	0	0
Ovh_{EAGC-q}	0	$q.\ c_{sig}$
$Ovh_{verifier}$	$q.\ c_{verify}$	$q.\ c_{verify}$

4.7.4.1. Estimation des coûts

Faisons quelques hypothèses sur les paramètres qui entrent en jeu. On considère le scénario suivant :

1. *n* sera égal au nombre de serveurs utilisés au sein du réseau IMS (CSCF, HSS, …), on suppose que $n=50$.
2. *q*, le nombre de requêtes de demande d'état de révocation des certificats par jour dépendra de n, selon la relation $q=n.p$ où $p= 0,1$ est le pourcentage de certificats révoqués par jour.
3. Le nombre de certificats gérés par une seule AC dépend de l'architecture de PKI sélectionnée. Dans notre cas on a une seule CA (HSS) qui gère tous les certificats des serveurs IMS. Donc le nombre moyen k des serveurs à gérer est égal à n le nombre total des serveurs : $k= 50$.
4. La fréquence de mises à jour des informations de révocation est d'une par jour, d'où $f = 1$.
5. RSA, algorithme asymétrique de cryptographie à clé publique, est utilisé pour la signature électronique, donc sa longueur est $l_{sig} =2048\ bits$.
6. Le numéro de série du certificat occupe une taille de $l_{sn} = 20\ bits$.
7. Concernant les coûts d'une opération de génération et de vérification d'une signature électronique, on fera l'hypothèse que $C_{sig} = 1200\ ms$ et $C_{verify} = 2\ ms$. (cas d'un ordinateur équipé d'un processeur Pentium III 750MHz et de 256M de mémoire vive).

Les coûts en bande passantes ainsi qu'en calcul sont donnés dans les deux tableaux suivants :

Tableau 4.8. Coût de mise à jour

Méthode de révocation	Bande passante entre l'AC et l'EAGC	Calcul au niveau de l'AC	Calcul au niveau de l'EAGC
CRL	20298 octet	1200 ms	2 ms
OCSP	0	0	0

Tableau 4.9. Coût de vérification

Méthode de révocation	Bande passante entre l'AC et l'EAGC	Calcul au niveau de l'AC	Calcul au niveau de l'EAGC	Calcul au niveau des vérificateurs
CRL	1024000 octet	0	0	10 ms
OCSP	10240 octet	0	6000 ms	10 ms

A partir de ces deux tableaux ainsi que la comparaison faite au niveau de [Cbek04], tenant en comptes les autres critères il est clair que :

OCSP : Cette méthode est consommatrice en calcul du fait que chaque requête de certificat parvenant au serveur suppose un message signé en retour. Par contre, la consommation en bande passante est faible du fait des messages courts échangés entre serveur OCSP et vérificateur. Mais comme le serveur OCSP est en ligne, un risque système important est induit.

CRL : Les avantages de la CRL sont sa simplicité, sa richesse en information et son faible risque système. Toutefois, la taille de la CRL constitue son inconvénient majeur, car la bande passante nécessaire à la mise à jour et à la vérification est très élevée, ce qui limite considérablement son extensibilité. Pour garantir sa fraîcheur, la CRL contient la date de la prochaine mise à jour de la CRL. De ce fait, les vérificateurs qui ont besoin d'informations de révocation fraîches vont tous au même moment vouloir récupérer la nouvelle CRL. Cela risque donc de provoquer une explosion du nombre de requêtes CRLs.

A travers les résultats obtenus, il semblerait, dans le cas d'un réseau d'opérateur, qu'un modèle d'infrastructure à clés publique à autorité de certification utilisant CRL pour la vérification et la révocation des certificats soit la solution à privilégier. En effet, le seul inconvénient et la consommation de la bande passante, mais comme le montre les deux tableaux de calcul cela ne dépasse pas le *1Mo* dans le meilleurs de cas et pour une fréquence de mise à jour d'une fois par jour.

4.8. Conclusion

Dans ce chapitre, nous avons proposé un nouveau protocole d'authentification et d'accord de clés (*Authentication and key agreement*) dans le réseau IMS. Ce protocole basé sur une infrastructure PKI permet de résoudre les problèmes de sécurité dont l'IMS-AKA souffre, conserve le *one-way* et améliore les échanges en termes de performance (par rapport à l'IMS-AKA).

Sur la base des résultats obtenues, nous pouvons confirmer que :

1) En termes d'échange de messages SIP/Cx, notre protocole peut économiser au moins 21,5 % par rapport à l'IMS-AKA.
2) En termes de consommation de la bande passante, le protocole proposé permet de réduire son utilisation de 27% par rapport à l'IMS-AKA.
3) En termes de sécurité, notre protocole résiste aux attaques atteignant la confidentialité et l'intégrité des données lors d'un enregistrement IMS (validé par AVISPA), contrairement à l'IMS-AKA qui n'assure ces deux propriétés que partiellement (seulement les messages de la deuxième phase qui sont protégés). En plus grâce au protocole proposé la manipulation des données au niveau du réseau de cœur n'est plus possible.
4) Notre protocole réduit le temps de réponses des requêtes d'enregistrement IMS.

Le tableau 4.10 présente une comparaison des différents protocoles discutés dans ce chapitre.

Tableau 4.10. Comparaison entre le protocole proposé et d'autres approches discutées dans ce chapitre

	IMS -AKA	One-Pass Authentication (Lin et al.)	Protocole Proposé Version I (DH)	Protocole Proposé Version II (PKI)
One-way	Non	Oui	Oui	Oui
Authentification Mutuelle	Oui	Non	Oui	Oui
Key agreement	Oui	Non	Oui	Oui
Confidentialité	Oui*	Non	Oui	Oui
Intégrité	Oui*	Non	Oui	Oui
Manipulation des données	Oui	Oui	Partiellement	Non
Non-répudiation au niveau du réseau de cœur IMS	Non	Non	Non	Oui
Niveau d'efficacité	---	Bon	Bon	Bon

*** seulement dans la seconde phase.**

Chapitre 5: Virtual Walled-Garden Model for IMS Services Provisioning

L'architecture IMS est la seule architecture de référence pour la convergence fixe/mobile actuellement normalisée. Avec la technologie IMS, un seul terminal serait en mesure d'être utilisé pour accéder à internet, regarder la télévision et en même temps servir de téléphone en utilisant un seul protocole de communication. Toutefois, l'architecture a été définie selon un modèle spécifique, le modèle que l'on appelle "*walled-garden*". Aujourd'hui, deux modèles de fourniture du service se font face. Celui déjà cité et le modèle appelé "*open-garden*" [ABGM09].

Dans un modèle fermé (walled-garden), les services sont fournis aux utilisateurs du même opérateur afin que les utilisateurs n'aient pas à chercher des applications en dehors du réseau IMS. Ces applications sont hébergées par l'opérateur de réseau IMS, qui garde le contrôle total sur les utilisateurs. Cependant, cette façon de faire est très restrictive pour les utilisateurs, car ils n'ont pas le choix de choisir les applications qu'ils souhaitent souscrire. Ils sont en effet retenus à ce qui est offert par leur opérateur de télécommunications.

La seconde approche est connue sous le nom du "open garden" et elle permet aux utilisateurs d'accéder à tout type d'applications hébergées par des fournisseurs de services externes. Les avantages d'utiliser un tiers fournisseur de services tiers sont essentiellement liés à la satisfaction de l'utilisateur, car tous les abonnés IMS auront pleinement accès à toutes sortes d'applications qui sont disponibles via l'Internet. En plus, les services externes évoluent à des vitesses Internet pour répondre aux demandes des clients. Néanmoins, ces services externes ne sont souvent pas de confiance et en conséquence ont rarement accès au profil complet des clients.

En IMS, les utilisateurs sont authentifiés avec le protocole IMS-AKA. Une fois l'authentification a réussi, le client aura accès complet à toutes les applications offertes par le réseau IMS mais cela n'est vrai que dans un contexte fermé (walled-garden). Alors que dans le second modèle, l'utilisateur devra s'authentifier à nouveau auprès de tous les serveurs externes

offrant un service, ce qui conduit à une augmentation du nombre d'authentifications effectuées lors d'une session d'application. Par conséquent, un mécanisme devrait être déployé pour permettre à l'utilisateur d'accéder à toutes leurs applications même les externes de manière transparente, comme dans le modèle fermé.

Dans ce chapitre, nous proposons un nouveau modèle de fourniture de services, ce que nous appelons *virtual walled-garden* [MKFO13]. L'objectif est de créer un univers autonome dans lequel les abonnés sont autorisés à profiter de tous les services et contenus offerts non seulement par leur opérateur, mais aussi par d'autres fournisseurs de services (SP : Services Providers), dans un environnement entièrement sécurisé et avec une expérience utilisateur et une qualité du service assurées. Pour relever ces défis, nous proposons une extension du modèle IMS existant pour accéder aux applications IMS qui sont situées en dehors du domaine IMS et entretenus par d'autres opérateurs de services. Ce modèle permettra de créer un lien de confiance entre le domaine IMS et services externes, et permettra de réduire la charge des utilisateurs finaux ainsi que les SPs par l'utilisation d'un mécanisme SSO multi niveaux (*Multi-Level Single Sign-On : ML-SSO*), réalisé par la fédération d'identité.

Ce chapitre sera devisé en quatre sous sections. Nous commençons par une description de l'architecture et les principes pour la mise en œuvre des services en IMS. Puis nous discutons les défis de fourniture de services en IMS. Dans la section 5.3 nous présentons un état de l'art des mécanismes de gestion d'identité (SSO "*Single Sign On*"). Notre seconde contribution le "*Virtual walled-garden*" fera l'objet de la section 5.4.

5.1. Architecture et principes pour la mise en œuvre des services

Les services IMS sont exécutés au niveau des serveurs d'application, ceux-ci pouvant appartenir au réseau de l'opérateur nominal (abonnement IMS de l'utilisateur) ou à un réseau externe (réseau visité en situation de *roaming* ou réseau tiers). Les serveurs d'application se trouvent à la bordure du domaine IMS afin de rendre les évolutions des services indépendantes des évolutions du réseau IMS (par exemple, les évolutions logicielles des CSCF) et de minimiser ainsi le délai de développement et lancement commercial des services.

Afin de permettre l'exécution de services de natures différentes, l'architecture IMS prévoit une interface unique depuis le S-CSCF vers les serveurs d'application comme illustré dans la figure 5.1.

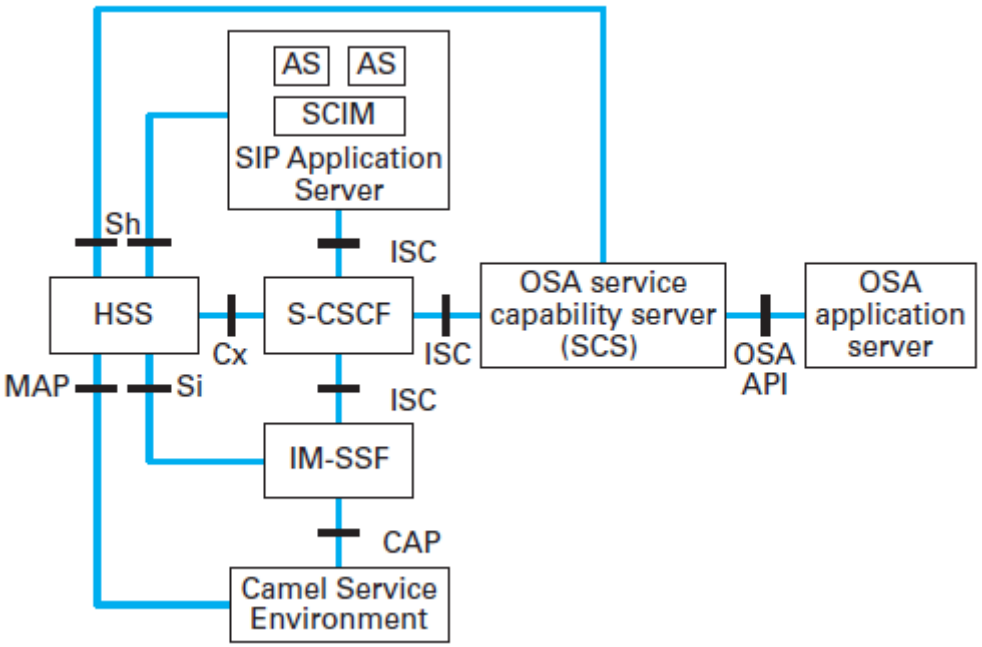


Figure 5.1. Architecture de référence pour le support des services IMS

Une adaptation est nécessaire dans le cas de OSA et CAMEL puisque cette interface ISC supporte seulement le protocole SIP. Le support des services de type OSA nécessite un serveur spécifique « *OSA service capability server* » utilisant une API OSA vers le serveur d'application OSA. Les services CAMEL sont supportés via l'entité « IM-SSF » qui réalise la conversion protocolaire entre SIP et le protocole CAP utilisé par les serveurs d'application CAMEL. Un serveur d'application peut intégrer une fonction de gestion de l'interaction des services, le « service capability interaction manager » (SCIM) non standardisée.

Les serveurs d'application s'interfacent donc avec :

– le S-CSCF via l'interface « ISC » basée sur SIP afin de déclencher les services hébergés sur les serveurs d'application ;
– le HSS via l'interface « Sh » basée sur Diameter (interface « Si » pour CAMEL) afin de récupérer les données du profil d'abonné IMS si nécessaire ;
– le terminal de l'utilisateur via l'interface « Ut » basé sur le protocole XCAP pour permettre à l'utilisateur la configuration de ses services.

5.1.1. Déclenchement des services sur la base de filtres

Le déclenchement des services repose sur l'usage de filtres (dits « *filter criteria* ») définis dans le profil de service de l'abonné IMS. Ces filtres contiennent notamment l'adresse du serveur d'application à contacter depuis le S-CSCF, les conditions de déclenchement vers l'AS (par exemple, une requête SIP « INVITE » pour un serveur d'application utile pour chaque appel de l'abonné), l'ordre dans lequel le serveur d'application doit intervenir pour le cas d'interaction de services (déclenchements successifs vers plusieurs serveurs d'application).

Les interactions de services se font via le S-CSCF selon le standard mais un serveur d'application peut selon sa propre logique de service interagir avec d'autres serveurs (en utilisant par exemple la fonction SCIM).

5.1.2. Identité des services

Les services IMS sont adressés dans l'IMS via des identités publiques de service. À l'instar des identités publiques d'utilisateurs, les identités publiques de service se présentent sous la forme de SIP URL ou Tel URL permettant d'assurer leur routage dans le domaine IMS. Elles identifient non plus des abonnés au service IMS, mais des services hébergés sur des serveurs d'application.

Les identités publiques de service présentent un intérêt fort pour l'usage de listes de diffusion comme par exemple des listes pour la messagerie instantanée ou pour la souscription à un service de présence. Il est possible pour les utilisateurs IMS de créer, gérer et utiliser des listes sous le contrôle des serveurs d'application concernés.

5.1.3. Configuration des services

L'utilisateur peut configurer ses services via plusieurs moyens et notamment par un accès direct aux serveurs d'application hébergeant ses services via l'interface « Ut » entre eux. L'interface « Ut » supporte le protocole XCAP basé sur HTTP. La configuration est matérialisée par des documents XML particuliers standardisés comme par exemple « application/resource-lists+xml » pour les listes. Il est possible de configurer les services en utilisant une requête SIP « INVITE » avec des paramètres spécifiques notamment pour la configuration des services supplémentaires (release 8). D'autres moyens non standardisés ne

sont pas exclus tels que la configuration via un serveur web ou une pré-configuration locale par l'opérateur. La figure 5.2 illustre un exemple de configuration de liste via l'interface « Ut ».

– L'utilisateur crée une liste via une requête XCAP « PUT » indiquant la nature du document et l'endroit où la stocker, et contenant le document XML de description de la liste à créer.

– Après vérification d'autorisation, le serveur accepte la création de la liste et y répond avec un message « 201 Created ». L'utilisateur peut ensuite ajouter (respectivement supprimer) des personnes à la liste avec une requête XCAP « PUT » (respectivement « DELETE ») que le serveur accepte avec une réponse « 200 OK ».

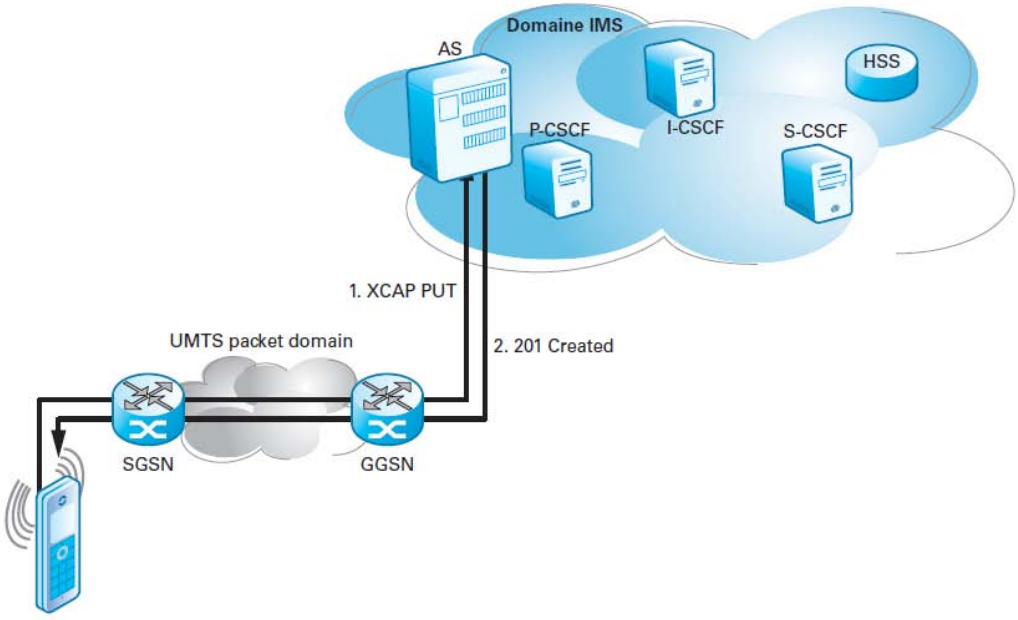


Figure 5.2. Configuration des services IMS via l'interface « Ut » entre le terminal et le serveur d'application

Que ce soit sur le modèle client/serveur avec un serveur d'application ou le modèle client/client avec un autre utilisateur, de nombreux exemples de services pourraient être cités tels que l'accès à des services web pour faire du commerce en ligne, télécharger des documents, faire du télé-enseignement etc., mais aussi l'entrée en conférence avec un grand nombre d'utilisateurs et l'échange de flux d'information divers avec l'ensemble des utilisateurs (par exemple pour partager un fichier de travail).

Il est à noter que pour certifier l'identité de l'utilisateur dans le domaine IMS, Le 3GPP réutilise les mécanismes définis par l'IETF (extension SIP). Cette extension définit deux nouveaux en-têtes SIP et un nouveau type d'anonymat (RFC 3325) :

– en-tête "*P-Preferred-Identity*" : l'utilisateur peut indiquer son identité préférée dans l'en-tête "*P-Preferred-Identity*" s'il souhaite être identifié auprès de son correspondant avec une identité publique particulière ;

– en-tête "*P-Asserted Identity*" : le P-CSCF insère une identité publique de l'utilisateur certifiée, c'est-à-dire l'une des identités publiques correspondant à l'enregistrement de l'utilisateur dans l'IMS. Le P-CSCF utilise l'identité préférée de l'utilisateur si l'en-tête présent "*P-Preferred-Identity*" est présent et si cette identité a fait l'objet d'un enregistrement (explicite ou implicite). Dans le cas contraire, le P-CSCF remplit l'en-tête "*P-Asserted Identity*" avec une identité enregistrée qu'il sélectionne par défaut ;

– type d'anonymat dit "*id*" : ce type d'anonymat exclut la communication de l'en-tête "*P-Asserted Identity*" en dehors du domaine de confiance (c'est-à-dire que l'identité publique certifiée reste dans le domaine IMS de l'opérateur home et le réseau visité en cas de roaming, mais n'est pas communiquée à des tiers et en particulier pas au terminal utilisateur).

L'IETF définit plusieurs mécanismes d'anonymat (RFC 3323) :

– en-tête "*From*" avec la valeur « Anonymous » pour masquer l'identité de l'appelant. L'usage de "*From*" : « *Anonymous* » dans un message SIP ne garantit pas la confidentialité de l'utilisateur dès lors où d'autres en-têtes SIP peuvent révéler son identité par exemples « *Contact* », « *Reply-To* », « *Via* », « *Call-Info* », « *User-Agent* », « *Organization* », « *Server* », « *Subject* », « *Call-ID* », « *In-Reply-To* », « *Warning* » ;

– en-tête "*Privacy*" pour permettre différents niveaux de confidentialité selon sa valeur notamment :

• « Header » : confidentialité sur l'ensemble des en-têtes contenant des informations sur l'identité de l'utilisateur,

• « None » : pas de confidentialité requise.

5.1.4. Exemple d'établissement de session

Etablir une communication entre deux clients nécessite le passage des étapes suivantes [Miik09] :

5. Recherche de l'abonné appelé,
6. Vérification des autorisations d'accès,
7. Mise en relation entre les correspondants.

La méthode associée à l'établissement d'appel est INVITE avec le protocole de signalisation SIP. Voici l'un des scénarios possible lors de l'exécution de service d'appel (Figure 5.3) :

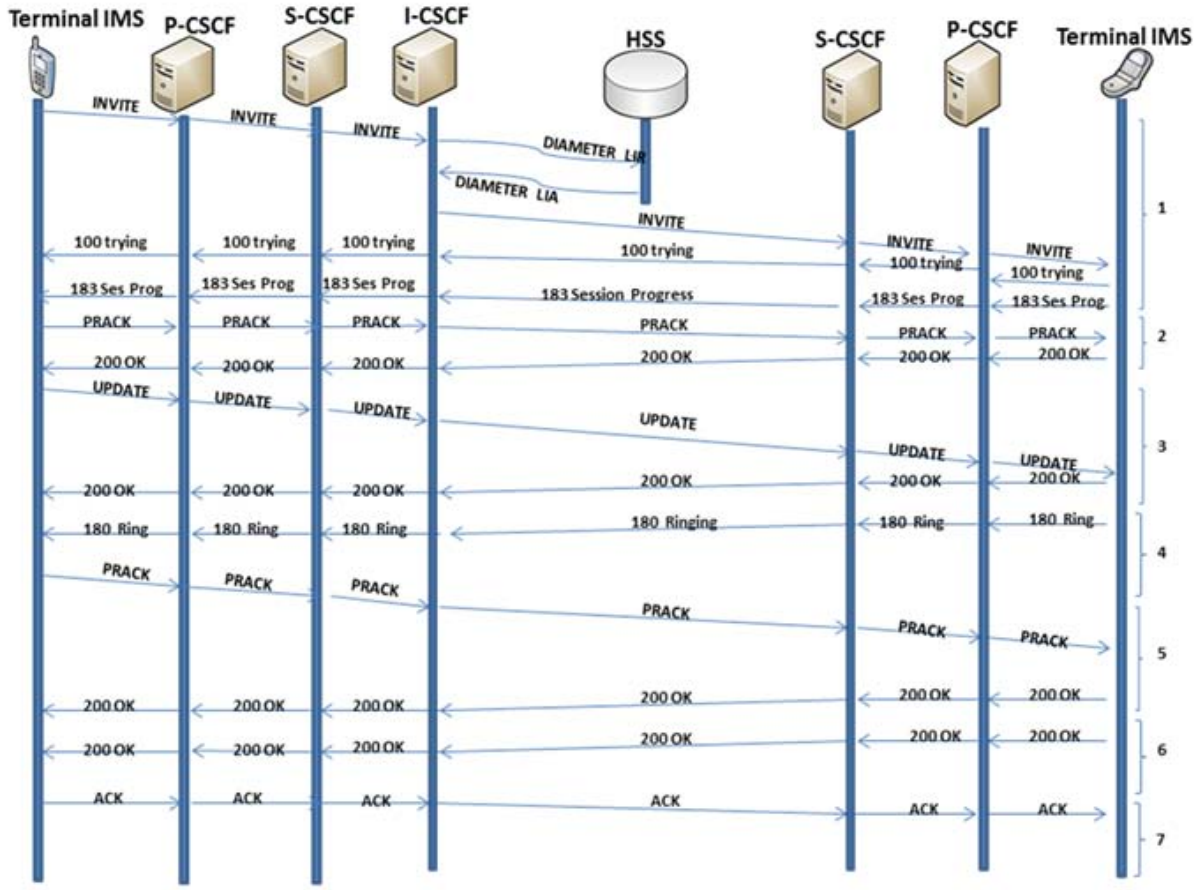


Figure 5.3. Mise en communication de deux terminaux

Dans ce scénario, nous allons supposer que l'appelant et l'appelé ont des opérateurs différents et soient localisés dans des réseaux visités, c'est-à-dire qui n'appartiennent pas forcement à leur opérateur respectif. C'est le cas le plus général.

Ce scénario de mise en relation de deux terminaux est découpé en sept étapes :

1. Un message d'invitation **INVITE** de A vers B, avec deux réponses temporaires : une réponse **100** pour indiquer la tentative et une réponse **183** pour négocier les paramètres de la communication. L'appelant doit s'adresser au serveur I-CSCF de B, qui le localise après une requête « **Diameter** ».

2. Pour s'assurer que l'émetteur A aie bien reçu la réponse **183**, il doit impérativement envoyer un acquittement temporaire.
3. Le terminal A doit négocier les paramètres de qualité de service pour garantir sa communication dans le réseau. Le terminal B vérifie que lui aussi a réservé les ressources nécessaires à la communication dans le réseau et valide la requête par sa réponse.
4. Le terminal de B commence à sonner. Cette étape complète les réponses temporaires à la requête d'invitation par une réponse **180**, elle aussi temporaire.
5. Pour s'assurer que cette réponse est bien reçue du terminal A, ce dernier doit confirmer la réception par une requête d'acquittement, qui attend elle-même une réponse.
6. Dès que l'utilisateur du terminal B répond, la réponse définitive 200 est envoyée à la requête initiale d'invitation.
7. La requête d'acquittement finale valide l'initialisation de la communication, qui peut dès lors débuter pour permettre aux terminaux de s'échanger des flux de données multimédias.

5.2. Défis de fourniture de services en IMS

L'idée maîtresse soutenant l'IMS est de fournir un cadre architectural unique pour la distribution des services dans des réseaux tout IP et quelle que soit la nature du réseau d'accès. L'IMS est donc avant tout une technologie qui supportera les services de télécommunications mais pas une implémentation en soi de ces services. L'IMS doit permettre de déployer tous les services tels que voix sur IP, présence, messagerie instantanée, Push to talk, conférence, distribution des services vidéo et de télévision.

La plupart des grands opérateurs de télécommunications planifient de déployer une infrastructure IMS dans le contexte du remplacement de leurs réseaux TDM. L'IMS doit apporter un effet de levier aux investissements nécessités par ce renouvellement de technologies en incluant notamment le support des services IPTV dans la même infrastructure de réseau que les services de voix sur IP. En effet, IMS définit une interface standard entre les serveurs d'application et le cœur de routage de la signalisation. Les serveurs d'application sont d'une part libérés des tâches de réservation des ressources au niveau des fonctions de contrôle du transport. D'autre part, ils vont partager un plan de contrôle commun facilitant la gestion des interactions entre différentes applications. En outre, IMS est par nature indépendant du

réseau d'accès, ce qui implique que l'utilisateur peut accéder à ses services à partir de différents types d'accès. De plus, cette capacité multi-accès est un premier support à l'implémentation de fonctions pour la continuité de service en cas de changement de réseau d'accès pendant la consommation du service.

Grâce à tous ces avantages, les opérateurs et les fournisseurs de services étaient au départ très enthousiastes pour déployer l'architecture IMS puisqu'il est prévu qu'avec IMS le revenu moyen par utilisateur (ARPU) pourrait augmenter de manière significative [IsGr09]. Toutefois, le taux de déploiement d'IMS a ralenti considérablement. Cette régression est due justement à l'utilisation de l'IMS dans le cadre d'un modèle fermé (walled-garden) avec un opérateur unique qui contrôle le réseau d'accès, réseau de cœur IMS et les serveurs d'application (AS).

Le but de ces opérateurs qui ont choisi d'adopter le modèle walled-garden est de créer un univers autonome dans lequel les abonnés sont autorisés à profiter de tous les services et le contenu offerts par leur opérateur, dans un environnement entièrement sécurisé et avec une expérience utilisateur et une qualité de service assurées.

Un autre avantage de ce modèle est l'assurance que les abonnés n'auront jamais avoir à chercher des contenus ou des services à l'extérieur du réseau, et, par conséquent, tous les revenus générés vont directement à l'opérateur. Malheureusement, ce modèle limite les utilisateurs finaux seulement aux services que leurs opérateurs IMS offrent. En outre, les services internes sont coûteux et prennent beaucoup de temps pour les développer, puisqu'il est très difficile chaque jour pour les opérateurs de développer de nouveaux services (par exemple la messagerie instantanée, réseaux sociaux). Par conséquent, le modèle walled-garden ne parvient pas à créer une demande massive des utilisateurs, qui est la principale force motrice du chiffre d'affaire.

Le freinage du déploiement de l'IMS peut être est dû à une autre limitation, il s'agit des multiples authentifications/autorisations. Dans l'architecture IMS d'aujourd'hui, l'utilisateur doit remplir au moins deux étapes d'authentification avant de recevoir des services IMS. L'utilisateur est authentifié premièrement par le réseau d'accès, ensuite, en utilisant IMS-AKA, par le réseau IMS. Si les services sont gérés par le même opérateur, aucune authentification n'est requise. Toutefois, pour bénéficier des services de tiers, l'utilisateur devra se ré-authentifier et passer les contrôle d'accès à nouveau auprès de chaque fournisseur de services.

Dans un marché compétitif, les utilisateurs aiment profiter de la liberté d'utiliser les services de tout fournisseur de contenu en fonction de leurs besoins et préférences. Pour attirer les opérateurs et prestataires de services, IMS doit démontrer qu'il est en effet une architecture multi-service qui peut être utilisé comme un cadre de services commun, même pour les services non-SIP, et certainement au moins pour les services Web. En fait, l'IMS a été dès le début conçu pour permettre la signalisation SIP de bout en bout entre IMS et les paramètres non-IMS, et si le point d'extrémité non-IMS ne prend pas en charge SIP, l'architecture de service IMS permet une intégration facile de passerelles de protocole. 3GPP a toujours eu l'intention de garder IMS ouverte aux réseaux non-IMS, et plus particulièrement à l'Internet. La création de nouveaux modèles fermés n'est pas une stratégie qui est durable pour les opérateurs dans les années à venir. Compte tenu de la prolifération des services d'Internet et de l'Internet, le succès éventuel de l'IMS serait proportionnel au trafic généré entre IMS et l'Internet. Une architecture IMS avec un très faible trafic de/vers Internet serait une architecture IMS qui n'a pas réussi à fournir une valeur ajoutée aux utilisateurs finaux, et par la suite les utilisateurs peuvent préférer contourner IMS pour accéder directement aux services sur Internet. Une preuve qu'IMS est ouvert à l'intégration avec les réseaux non-IMS en particulier Internet est la dépendance entre l'organisation de deux normes, le 3GPP et l'IETF.

Par conséquent, l'ouverture des opérateurs IMS auprès de tiers fournisseurs de services tiers tend à être une obligation pour assurer le succès du réseau IMS. Les Services externes d'autre part se déplacent à des vitesses Internet appropriées pour répondre aux demandes des clients. Néanmoins, ces services externes ne sont souvent pas de confiance et en conséquence ont rarement accès au profil complet de la clientèle. Pour relever ces défis, nous proposons une extension du modèle IMS existantes pour accéder aux applications IMS qui sont situées en dehors du domaine IMS et entretenus par d'autres opérateurs de services, ce que nous appelions "virtual walled-garden model" ou modèle muré/fermé virtuel. Ce modèle permettra de créer un lien de confiance entre le domaine IMS et les services externes, et permettra de réduire la charge des utilisateurs ainsi que les SPs par l'utilisation d'un mécanisme SSO en utilisant une fédération d'identité.

En fait, SSO est un bon moyen de fournir la facilité d'utilisation car il permet aux utilisateurs de sauter des processus d'authentification gênants lors des accès à de multiples services. Mais puisque l'utilisateur est authentifié seulement une fois avec une méthode d'authentification, il peut y avoir une dégradation de la sécurité. Par exemple pour l'accès aux banques et autres SPs

avec des exigences plus élevées de sécurité, le SSO ne peut pas être une bonne solution. À la suite de cette observation notre solution basée sur SSO ne devrait pas traiter tous les SPs de la même façon et avec le même niveau de sécurité. Une façon d'améliorer cette solution est d'introduire la notion de niveau de sécurité qui seront affectés aux SPs, mettant en œuvre ce qu'on peut appeler "Multi-level virtual walled-garden model" ou bien modèle muré virtuel multi-niveaux (Figure 5.4).

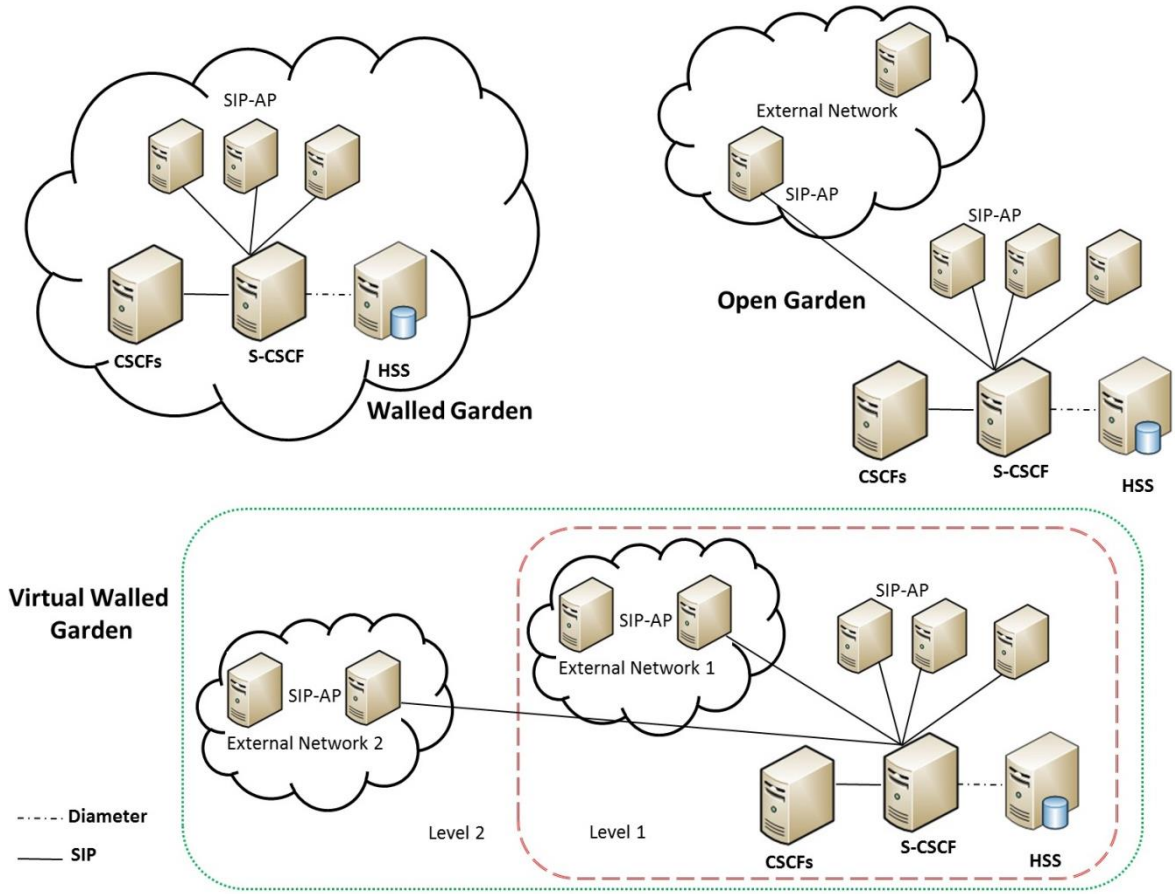


Figure 5.4. Le modèle Virtual Walled-Garden

5.3. Single Sign On : SSO

5.3.1. Qu'est-ce que le SSO ?

La multiplication des accès et des mots de passe amène les utilisateurs à faire de leur mieux pour conserver les informations relatives à leurs comptes. Malheureusement cela inclut souvent le recours à des pratiques dangereuses telles que : inscrire les codes secrets sur leur agenda papier ou sur des post-it, utiliser le même pour la plupart de leurs accès, ou laisser les connexions ouvertes lorsqu'ils quittent leur poste de travail. Une bonne solution pour ce problème est de s'appuyer sur une solution de Single Sign-On. Le Single Sign-On (SSO) est

un processus qui permet à un utilisateur de s'authentifier une seule fois pour accéder à plusieurs applications ou ressources.

Mais aussi pratiques soient-ils, les SSO sont rarement déployés simplement pour faciliter la vie des utilisateurs. Ils s'intègrent généralement à un projet de sécurité plus large, dans lequel ils sont considérés comme un élément secondaire.

5.3.2. Objectifs d'un système de Single Sign-On

5.3.2.1. Apport ergonomique pour l'utilisateur

L'atout évident d'un service de SSO est la simplification des procédures d'authentification pour l'utilisateur. Alors qu'il devait s'identifier successivement auprès de chaque application lors d'une session de travail, le SSO lui permet de ne s'authentifier qu'une seule fois. L'identifiant de l'utilisateur et ses attributs sont ensuite propagés vers les applications. Certains logiciels de SSO assurent également la fermeture de toutes les sessions applicatives de l'utilisateur lorsqu'il se déconnecte.

Ce besoin de cohérence des systèmes d'authentification dans les applications web de l'établissement est renforcé dans le cadre du déploiement de portails web d'établissement. En effet ceux-ci visent à présenter tous les outils mis à disposition de l'utilisateur de façon homogène et cohérente, alors que les applications sont très hétérogènes.

5.3.2.2. Amélioration de la sécurité

L'accès nomade par le web à des applications nécessite de sécuriser la phase d'authentification sans faire d'hypothèse sur la sécurité du réseau qui relie clients et serveurs. L'architecture de type SSO, en concentrant cet effort de sécurisation sur le (ou les) serveur(s) d'authentification, permet de mettre en œuvre sur ce point une politique de sécurité cohérente tout en allégeant l'effort d'écriture des applications. L'utilisation d'un service commun d'authentification devrait également faciliter l'évolution des méthodes d'authentification (certificats X509, Kerberos,…) ou la prise en compte de plusieurs niveaux d'authentification en fonction de la sensibilité des applications accédées.

La mise en place d'un SSO est en outre l'occasion de donner de bonnes habitudes aux utilisateurs : ils ne doivent délivrer leur mot de passe qu'au seul serveur d'authentification

dont la bannière de login et l'URL (utilisation d'un certificat serveur recommandée) sont bien identifiés.

5.3.3. Les types d'authentification SSO

Il existe deux types d'authentification SSO; la première dite Web SSO, le seconde dite Enterprise SSO (eSSO).

Le Web SSO prend en charge toutes les applications qui utilisent un navigateur Web pour vous connecter à des applications (Figure 5.5).

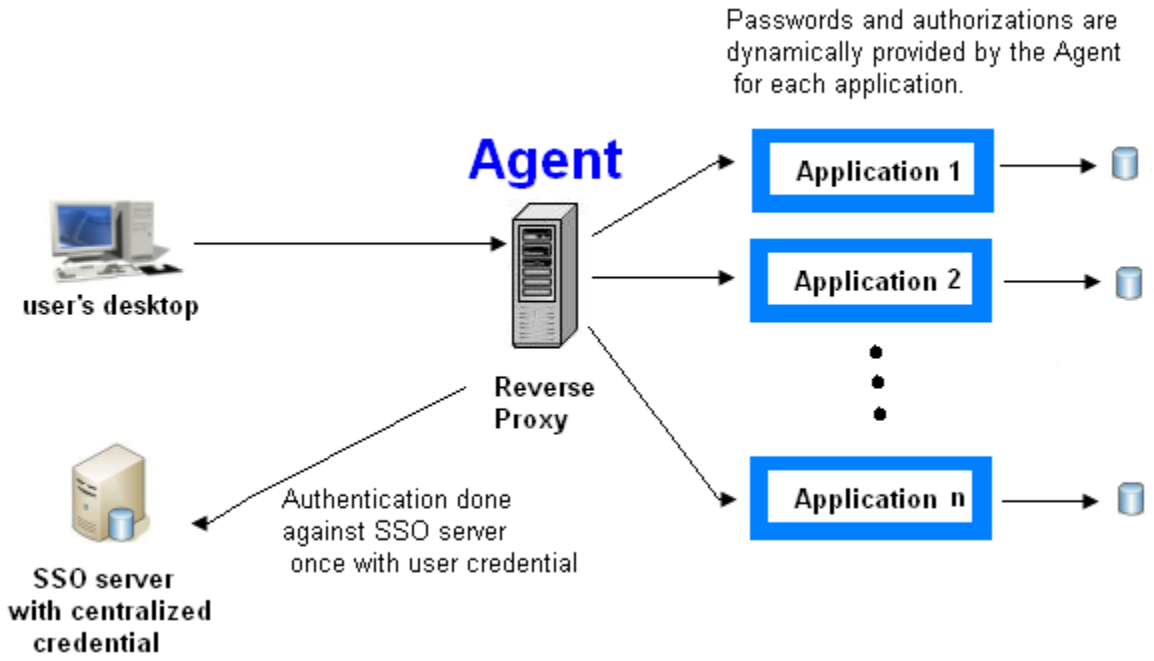


Figure 5.5. Architecture Web SSO

Pour leur part, les systèmes eSSO ne sont pas limités aux applications web et sont conçus pour minimiser le nombre de fois qu'un utilisateur doit taper son login et son mot de passe pour se connecter à de multiples applications de l'entreprise (Figure 5.6).

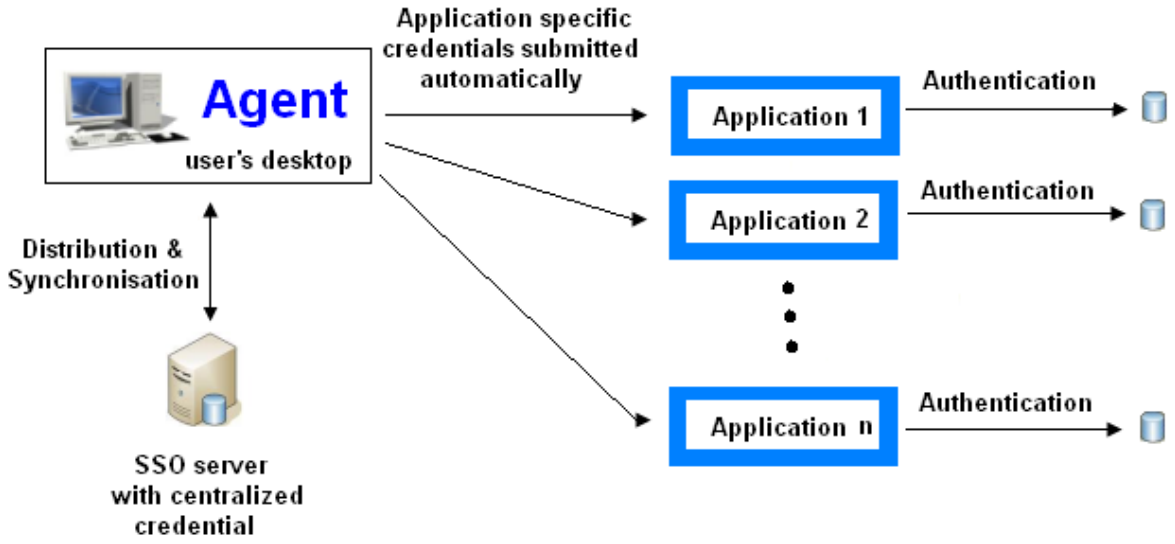


Figure 5.6. Enterprise SSO architecture

5.3.4. Les composantes SSO

Dans le système SSO, nous trouvons généralement les éléments suivants:

- Le client qui demande l'accès à l'application. Généralement, il s'agit d'un navigateur Web (cas d'un WebSSO). Dans le cas plus général d'une application client / serveur, le client peut être par exemple un client telnet.
- Le serveur d'authentification qui conserve la base de données des informations d'indentification. C'est l'élément central du système de SSO.
- Le serveur d'application qui fournit les ressources en fonction du résultat du processus d'authentification.
- L'agent SSO qui s'interpose entre le client et les serveurs d'applications prend en charge (au moins en partie) la phase d'authentification du client vis-à-vis du serveur d'application. L'agent peut être localisé à différents endroits selon l'architecture SSO, et peut être matériel ou logiciel.

La figure 5.7 présente l'échange de message classique entre les composantes SSO pour un établissement d'une authentification d'une application.

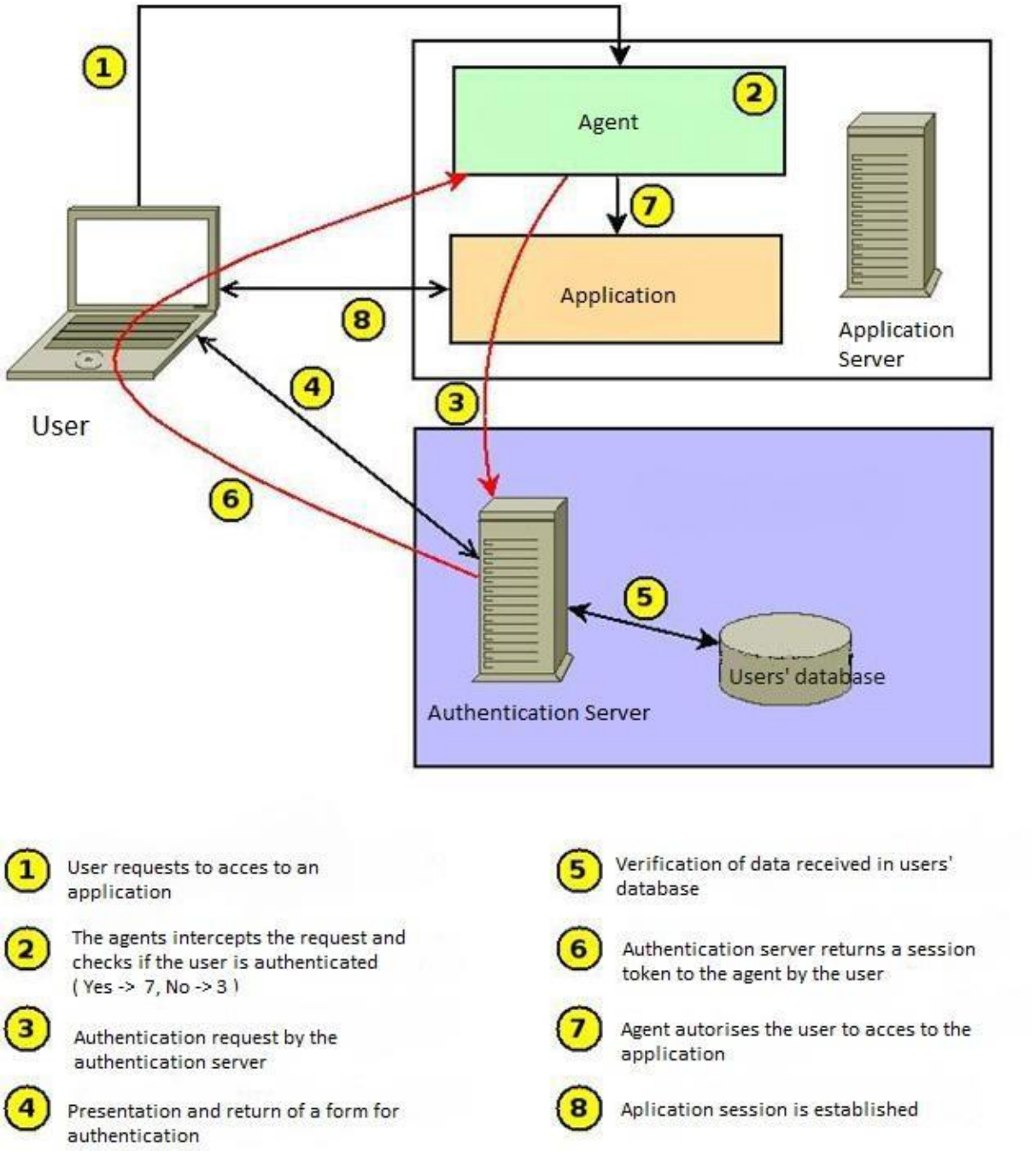


Figure 5.7. Etablissement d'une session pour une application utilisant SSO

5.3.5. Les différentes approches de SSO

5.3.5.1. L'approche centralisée

Le principe est de disposer d'une base de données centralisée contenant tous les utilisateurs. Cela permet également de centraliser la gestion de la politique de sécurité. Un exemple de mise en œuvre est LDAP (Figure 5.8).

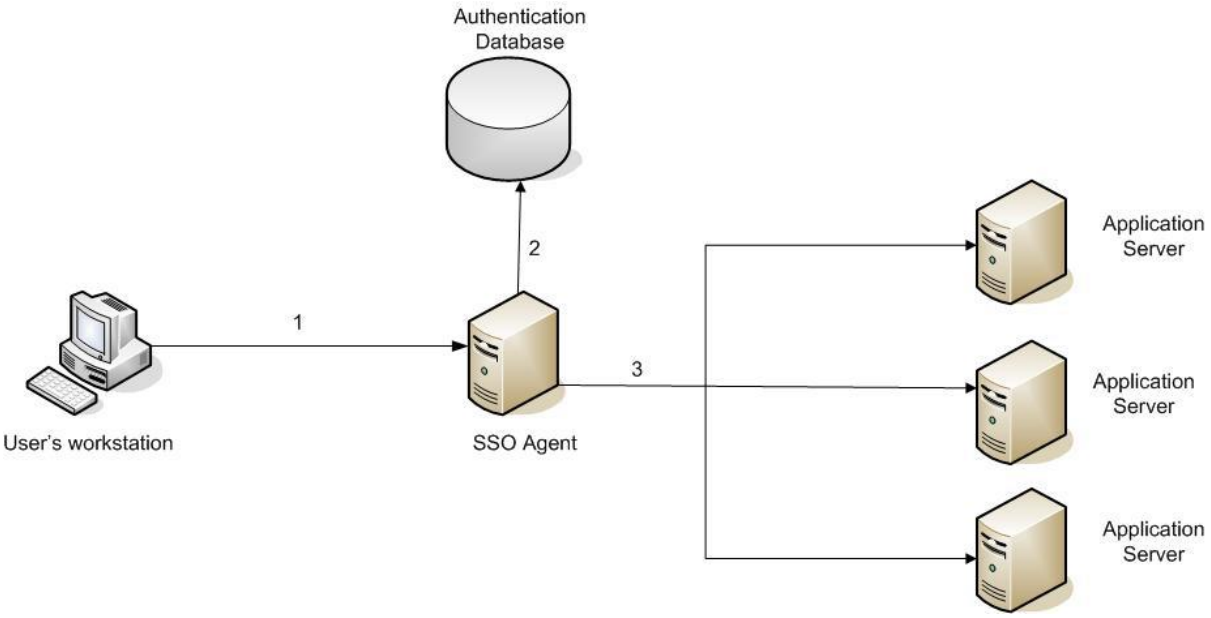


Figure 5.8. Approche SSO centralisée

1. Le client souhaite accéder à une application. Dans ce cas, l'agent s'exécute sur un Reverse Proxy et intercepte la demande.
2. L'agent authentifie l'utilisateur d'après une base de données d'authentification qui peut être un annuaire LDAP.
3. Une fois l'utilisateur authentifié, il peut accéder à l'application.

5.3.5.2. L'approche fédérative

Le grand défi dans les infrastructures d'authentification aujourd'hui est d'étendre le SSO pour couvrir plusieurs autorités d'authentification différentes (mise en œuvre sur différentes plates-formes ou gestion par des organisations différentes). La fédération permet d'étendre le contrôle d'accès et le SSO à travers les frontières organisationnelles. En effet, la mise en œuvre de l'identité fédérée et l'extension du SSO dans les entreprises permet de distribuer le contrôle et la maintenance des activités, et donc d'avoir plus de commodité et de temps à la fois pour les organisations et les utilisateurs. L'approche fédérée permet à un utilisateur de manière transparente de parcourir les différents sites, services au sein d'une fédération donnée. Chaque service gère une partie des données d'un utilisateur mais partage les informations de cet utilisateur avec les services partenaires.

Cette approche a été développée pour répondre à un besoin de gestion décentralisée des utilisateurs, où chaque service partenaire désire conserver le contrôle de sa politique de sécurité (Figure 5.9).

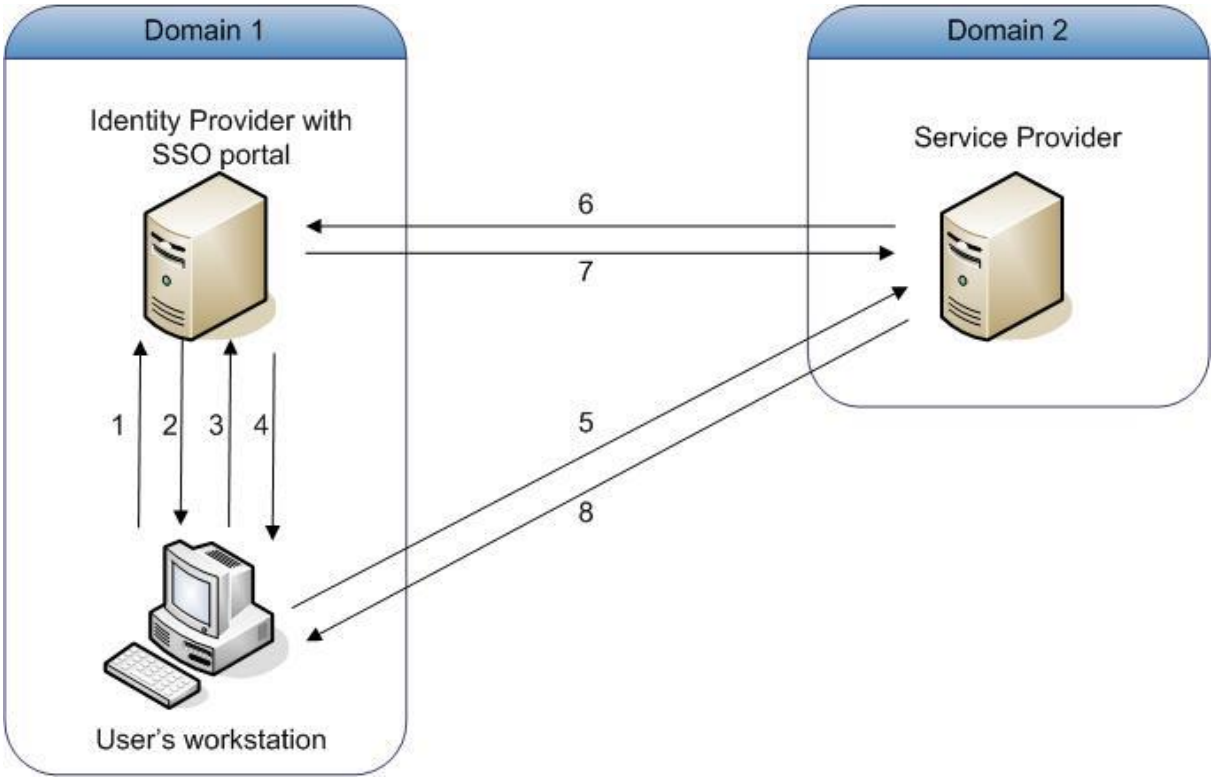


Figure 5.9. Approche SSO fédérative

1. L'utilisateur se connecte au fournisseur d'identité (IdP).
2. Après une authentification réussie, l'IdP envoie à l'utilisateur des informations sur les applications auxquelles il peut accéder.
3. L'utilisateur clique sur le lien Service Provider (SP) dans le portail. Il s'agit d'un lien spécial, qui ne se connecte pas directement au SP.
4. L'IdP reçoit la demande et crée une "Identity Assertion" (un identifiant d'identité). L'IdP conserve cette "Identity Assertion" avec un "artefact" pointeur dans son cache. Puis, l'IdP renvoie une réponse redirigée vers le navigateur client.
5. Le navigateur est redirigé vers le SP avec "l'artefact".
6. Le SP reçoit cette demande et contacte l'IdP avec "l'artefact" "pour demander "l'Identity Assertion" réelle.
7. L'IdP reçoit la demande, et vérifie cette entrée dans la table des "Identity Assertion" en cache en utilisant "l'artefact" comme index. Il crée une "Identity Assertion" au format SAML, et le renvoie à la SP.

8. Le SP extrait les informations utilisateur de "l'Identity Assertion" reçue. Enfin, après une authentification locale réussie, l'utilisateur est autorisé à accéder au service.

Le principal exemple de l'approche fédérée est Liberty Alliance. Elle s'appuie principalement sur la norme SAML, ainsi que sur les protocoles http et SSL.

Un autre protocole pour établir la confiance entre des systèmes hétérogènes d'identité existe. C'est un protocole relativement nouveau qui est appelé *Web Service Federation* (*WS-Federation*).

5.3.5.3. L'approche coopérative

Cette approche est similaire à l'approche de la Fédération. Elle répond aux besoins de structures institutionnelles, par exemple, des laboratoires de recherche ou des administrations. Dans l'approche coopérative, chaque utilisateur dépend de l'une des entités partenaires. Quand il essaie de parvenir à un service du réseau, l'utilisateur est authentifié par le partenaire dont il dépend. Comme dans l'approche fédérative, tous les services du réseau gèrent indépendamment leur propre politique de sécurité. Avec cette approche, les identifiants de sécurité de l'utilisateur ne sont pas échangés. Les principaux représentants de cette approche sont Shibboleth [Shib00].

5.3.6. OpenID et OAuth

5.3.6.1. OpenID

L'identification sur le Web se résume souvent à un couple nom d'utilisateur/mot de passe, email/mot de passe ou encore code client/mot de passe. Chaque site Web :

- a sa propre politique de nom d'utilisateur (avec ou sans caractères spéciaux etc.)
- a sa propre politique de mot de passe (longueur minimale, avec ou sans caractères spéciaux etc.)

On peut se retrouver facilement avec une bonne dizaine de combinaisons au fur et à mesure que l'on multiplie le nombre d'adhésions sur des sites Web. C'est d'autant plus le cas que l'on possède plusieurs identités :

- son identité personnelle

- son identité professionnelle
- une à plusieurs identités en fonction de ses centres d'intérêts (jeux, photo etc.)

OpenID [ReRe06a, ReRe06b] c'est un seul paramètre : une URL du genre *http://_pseudonyme_.serveur-openid.com*. Disposer d'une adresse OpenID c'est l'avantage de :

- gérer plusieurs identités à partir d'une seule adresse
- n'avoir plus qu'un seul mot de passe : celui de son adresse OpenID
- ne plus avoir à s'inscrire : votre URL est déjà votre identifiant
- ne plus galérer pour trouver un nom d'utilisateur alambiqué comme toto9898464 : votre URL est unique

En s'authentifiant sur un service compatible OpenID, l'identifiant et le mot de passe sont à fournir au prestataire OpenID (pour vérifier qu'il s'agit bien de vous), la dernière étape étant celle de l'autorisation du service. Le prestataire OpenID vous met en relation de confiance, relation qui peut s'arrêter du jour au lendemain. Vous révoquez ainsi l'accès du service à vos données personnelles.

5.3.6.2. OAuth

OAuth [Hard12] est un protocole ouvert permettant à une application d'accéder aux informations relatives à un utilisateur final auprès d'un service web lorsque l'application est autorisée par l'utilisateur final. Les informations relatives à l'utilisateur final sont transférées en toute sécurité sans révéler l'identité de l'utilisateur.

OAuth a pour objectif d'obtenir auprès du serveur web un jeton d'accès, qui peut ensuite être utilisé pour échanger des données propres à l'utilisateur avec un service web (par exemple des informations de calendrier ou un répertoire d'adresses). La procédure OAuth normale est une séquence à quatre étapes :

1. Demander un jeton de "demande".
2. Demander que le jeton soit autorisé, ce qui entraîne l'approbation par l'utilisateur.
3. Echanger le jeton de demande autorisé contre un jeton d' "accès".

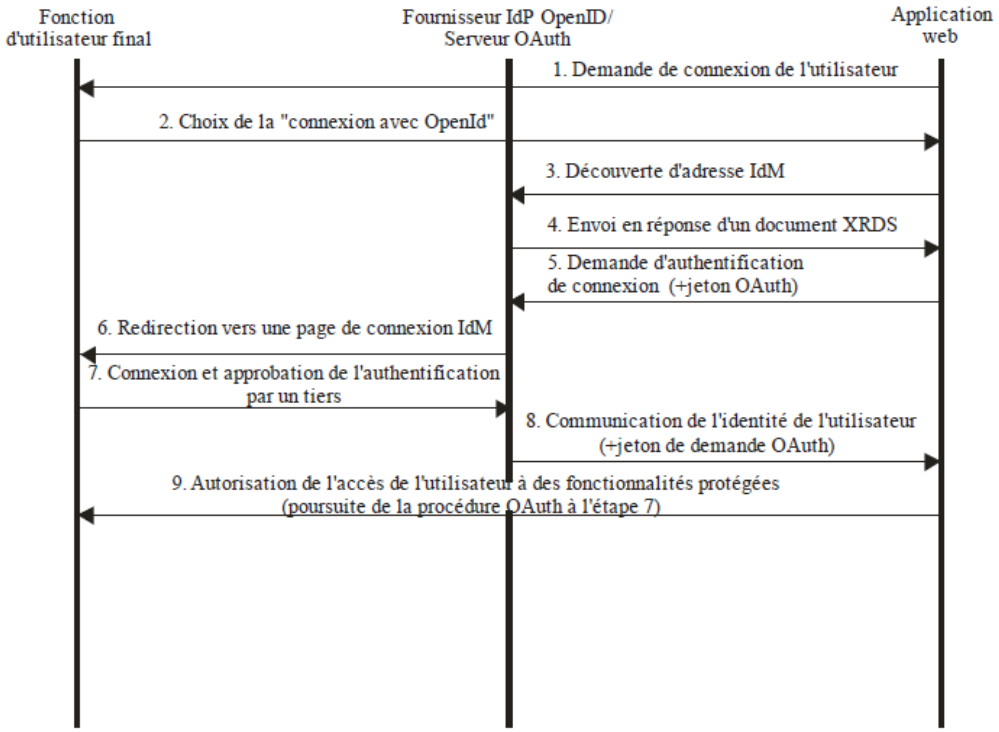
4. Utiliser le jeton d'accès pour interagir avec les données de service web relatives à l'utilisateur.

5.3.6.3. Utilisation de OpenID conjointement avec OAuth

Tandis que OpenID peut être utilisé comme mécanisme de gestion d'identité pour authentifier des utilisateurs, OAuth pourrait également être utilisé pour autoriser l'accès à des données d'utilisateur sensibles. Dans un tel scénario, le fournisseur IdM (Identity Management) assure des fonctions combinées et fait office à la fois de fournisseur d'identité OpenID et de fournisseur de service OAuth.

5.3.6.4. Flux d'autorisation OpenID + OAuth

Avec OpenID + OAuth, la séquence reste essentiellement la même. La différence est que l'obtention d'un jeton de demande OAuth autorisé (étapes 1 et 2 de la figure 5.10) est incluse dans la demande d'authentification OpenID. De cette façon, l'utilisateur peut approuver simultanément une connexion et un accès aux services.



XRDS : séquence de descripteurs de ressources extensible (*eXtensible Resource Descriptor Sequence*)

Figure 5.10. Authentification basée sur OpenID + OAuth

Les étapes de base sont les suivantes:

1. L'application web demande à l'utilisateur final de se connecter en lui offrant un ensemble d'options de connexion, y compris l'utilisation de son compte OpenID.
2. L'utilisateur choisit l'option de "connexion avec OpenID".
3. L'application web envoie une demande de "découverte" au fournisseur de service d'identité IdSP (Identity Service Provider) pour obtenir des informations relatives au point d'extrémité d'authentification de connexion IdSP.
4. Le fournisseur IdSP retourne un document XRDS (*eXtensible Resource Descriptor Sequence*), qui contient l'adresse du point d'extrémité.
5. L'application web envoie une demande d'authentification de connexion à l'adresse du point d'extrémité IdSP.
6. Cette action redirige l'utilisateur vers une page de connexion fédérée IdSP, soit dans la même fenêtre du navigateur soit dans une fenêtre incrustée, et il est demandé à l'utilisateur de se connecter.
7. Une fois l'utilisateur connecté, le fournisseur IdSP affiche une page de confirmation et indique à l'utilisateur qu'une application tierce demande une authentification. Sur la page, il est demandé à l'utilisateur de confirmer ou de rejeter l'établissement d'un lien entre la connexion au compte IdSP et la connexion à l'application web. Il est ensuite demandé à l'utilisateur d'approuver l'accès à un ensemble spécifié de services IdSP. Pour que l'authentification puisse se poursuivre, l'utilisateur doit approuver à la fois la connexion et le partage des informations le concernant.
8. Si l'utilisateur approuve l'authentification, le fournisseur IdSP retourne des informations relatives à l'utilisateur à l'URL spécifiée dans le paramètre openid.return_to de la demande initiale. Un identificateur fourni par le fournisseur IdSP, qui n'a aucun lien avec le nom ou le mot de passe associé au compte IdM actuel de l'utilisateur, est joint en tant que paramètre de requête openid.claimed_id. Si la demande portait aussi sur un échange d'attributs, d'autres informations relatives à l'utilisateur peuvent être jointes. Concernant OpenID + OAuth, un jeton de demande OAuth autorisé est également retourné.
9. L'application web utilise l'identificateur fourni par le fournisseur IdSP pour reconnaître l'utilisateur et autoriser l'accès aux fonctionnalités et aux données de l'application. Concernant OpenID+OAuth, l'application web utilise le jeton de demande pour poursuivre la séquence OAuth et obtenir un accès aux services IdSP de l'utilisateur.

5.3.7. SAML

SAML (Security Assertion Markup Language [Hloc08]) a été initialement conçu pour permettre entre autres la délégation d'authentification. C'est devenu un standard OASIS [Oasi00] en 2002. Il s'agit d'un ensemble de spécifications qui définissent comment des services peuvent s'échanger des assertions de sécurité (authentification, autorisation, attributs), indépendamment des technologies utilisées par chacun de ces services (PKI, SSO, LDAP, Kerberos, etc.). SAML ne couvre donc pas tout le spectre de la gestion des identités, par exemple ne définit pas de protocole de SSO ou une sémantique d'attributs standard. Il s'appuie sur des standards préexistants (XML, SSL, etc.) et a été conçu avec suffisamment d'abstraction pour rendre inter opérable des systèmes hétérogènes, et s'articuler au mieux avec d'autres mécanismes de gestion d'identités.

SAML est constitué de différents protocoles, qui correspondent aux différents cas d'usage adressés par ce standard. Un protocole SAML décrit de façon abstraite comment une entité interagit avec un système SAML, généralement sous la forme d'une séquence de requêtes et de réponses. Un "*protocol binding*" est la traduction d'un tel protocole abstrait en un protocole de communication implémentable informatiquement, par exemple sous la forme de Web Services SOAP [BEKL03, Suda03]. De plus, SAML étant très abstrait pour assurer l'interopérabilité des systèmes (notamment sur la composition des messages), il existe des "profils SAML" qui restreignent (ou étendent) la variabilité d'un protocole de base pour des usages particuliers. En s'accordant sur l'utilisation d'un certain profil, deux entités voulant communiquer en SAML se simplifient l'interopérabilité.

Normalisé, dans sa version 2.0, en mai 2005 par l'OASIS, SAML permet l'échange sécurisé d'informations d'identités (authentification et autorisation). SAML définit le format du message XML, appelé assertion, ainsi qu'un ensemble de profils. Ces profils sont des cas d'utilisation détaillés qui présentent la cinématique d'échange des messages, les paramètres attendus et renvoyés.

Dans sa version 2.0, normalisée en mai 2005 par l'OASIS, SAML définit deux briques essentielles pour sécuriser les échanges :

- Le SP (Service Provider), fournisseur de service, protège l'accès aux applications. Il refuse tout accès sans authentification préalable et redirige l'utilisateur non authentifié vers son fournisseur d'identité.
- L'IdP (Identity Provider), fournisseur d'identité, s'occupe d'authentifier l'utilisateur ainsi que de récupérer des informations additionnelles associées à son identité.

Ce mode de fonctionnement est suffisant pour une utilisation cantonnée à l'entreprise avec un annuaire des identités centralisé.

Dans le cadre d'une fédération entre plusieurs domaines d'identification, SAML définit une troisième brique appelée le DS (*Discovery Service*) qui permet à l'utilisateur de sélectionner manuellement son domaine parmi une liste. Avec un peu de configuration, il est possible de supprimer cet élément, un peu bloquant pour les utilisateurs.

5.3.7.1. Profiles SAML

Les normes SAML et Shibboleth (cette dernière est basée elle-même sur SAML) offrent plusieurs modes d'utilisation, nommés “profils”. Ils définissent comment sont utilisés les messages SAML pour un contexte donné : requêter une authentification, réaliser du SSO, déconnexion globale, etc.

Le profil le plus courant, appelé "Web Browser SSO", décrit, entre autres, les étapes d'authentification d'un utilisateur et les allers-retours entre le SP et l'IdP. L'utilisateur tente d'accéder à sa ressource protégée par le SP. Le SP vérifie que l'utilisateur est authentifié et s'il ne l'est pas, le redirige vers son IdP. L'IdP demande à l'utilisateur de s'authentifier (identifiant puis mot de passe par exemple) puis renvoie une assertion SAML au SP contenant l'identité de l'utilisateur et la garantie qu'il est authentifié. Le SP autorise alors l'utilisateur à accéder à la ressource initialement demandée.

Ce mécanisme d'authentification repose sur les redirections du navigateur Internet. Ce profil permet aussi de récupérer un ensemble d'attributs supplémentaires liés à l'identité de l'utilisateur et demandés par la ressource.

Un second profil basé sur des artéfacts, permet de dé-corréler l'authentification de la récupération des informations d'identité de l'utilisateur. Le SP reçoit de l'IdP, par le navigateur

Internet de l'utilisateur, une assertion SAML contentant un artefact. Le SP doit alors interroger directement l'IdP pour obtenir les informations liées à l'identité de l'utilisateur.

D'autres profils décrivent comment mettre en œuvre le DS, les notions de "*logout*" et la possibilité de se passer du navigateur de l'utilisateur pour transmettre les assertions SAML entre services.

5.3.7.2. Sécurité en SAML

Les assertions SAML sont basées sur les couches SOAP, XML Encryption et XML Signature.

- SOAP est le protocole d'encapsulation standard des messages XML, utilisé principalement par les Web services.
- XML Encryption est le protocole standard de chiffrement des messages XML. Il a la particularité de pouvoir chiffrer la globalité du message ou simplement un sous-ensemble précis. Cela permet d'avoir par exemple un document XML en clair avec des valeurs d'attributs chiffrées.
- XML Signature est le protocole standard de signature des messages XML. Tout comme XML Encryption il permet de cibler l'élément à signer. Cela permet à plusieurs intervenants de signer chacun une partie différente du document XML.

Le SP et L'IdP sont deux entités qui ont connaissance chacune l'une de l'autre en termes d'identifiant et de certificat. Les messages XML qui transitent sur le réseau sont donc chiffrés par la clé publique du destinataire, seul capable de déchiffrer le message avec sa clé privée. L'émetteur signe ses assertions avec sa clé privée permettant au destinataire de vérifier sa provenance.

5.4. Virtual walled-garden

Nous avons vu que le modèle fermé est un facteur limitatif pour l'utilisateur et que l'ouverture des opérateurs IMS auprès de tiers fournisseurs de services tiers tend à être une obligation pour assurer le succès du réseau IMS. Ces fournisseurs de services tiers cependant n'accorderont l'accès qu'aux utilisateurs authentifiés.

L'objectif d'intégrer SSO dans un environnement IMS est de permettre à l'utilisateur d'accéder à ces services de tiers sans avoir à s'authentifier de nouveau à chaque fois.

Dans ce qui suit, nous expliquons comment il est possible d'étendre l'architecture IMS, afin de permettre aux utilisateurs d'atteindre les différents services au-delà même du domaine de leur opérateur IMS.

5.4.1. Approche SSO choisie

Nous pouvons éliminer immédiatement l'approche SSO centralisée dans notre cas pour de nombreuses raisons. En effet nous avons besoin d'un système fonctionnant dans un contexte multi-domaines, le domaine de l'opérateur IMS et les domaines des prestataires de services. De plus l'approche SSO centralisée permet l'authentification à des services différents avec une seule identité, ce qui peut être une issue pour la garantie de la vie privée. Enfin, les systèmes centralisés ne permettent pas la transmission d'attributs d'autorisation ou d'information de l'utilisateur.

Donc, nous devons concentrer notre attention sur les deux autres approches, qui permettent à la fois d'accéder aux services de plusieurs domaines et de protéger l'identité de l'utilisateur.

Dans la solution présentée, nous pouvons trouver des caractéristiques des deux approches fédérative et coopérative :

- Les utilisateurs disposent de plusieurs identités et des comptes, un dans chaque domaine, avec des informations de profil distribuées.
  En effet, il est intéressant pour les différents fournisseurs de services de conserver des informations sur chaque utilisateur, des informations qui sont en général spécifiques à l'application, par exemple le crédit restant de l'utilisateur qui sera utilisé au cours de la phase d'autorisation et ne devraient pas être géré par un seul IdP.
  De ce point de vue, la solution pourrait être considérée comme fédérative.
- Ces différentes identités sont fédérées avec l'identité IMS. L'identité de l'utilisateur IMS est fédérée avec son identité correspondant à chaque fournisseur de services.
  Ceci permet à l'utilisateur d'accéder à tous les fournisseurs de service dès qu'il est authentifié avec son identité IMS, mais ne permet pas à un utilisateur de s'authentifier auprès d'un fournisseur de service spécifique pour accéder à un autre fournisseur de services sans nouvelle authentification.
  De ce point de vue, la solution pourrait être considéré comme à la fois fédérative et coopérative.

- L'authentification est effectuée uniquement avec un seul IdP (situé dans le domaine de l'opérateur ou un fournisseur de l'identité tiers en s'appuyant sur l'authentification de l'opérateur).
  De ce point de vue, la solution pourrait être considérée comme coopérative.

La solution proposée est donc une solution hybride située entre coopération et fédération.

5.4.2. Intégration des mécanismes SSO en IMS

L'ajout des composants SSO dans l'architecture IMS ne devrait pas modifier son comportement natif et son objectif. En outre, les normes de la fédération d'identités existants, tels que le cadre de fédération Liberty Alliance ou Shibboleth sont limités à des services Web. C'est la raison pour laquelle nous devons adapter les mécanismes de fédération existants au mode IMS/SIP.

Comme expliqué ci-dessus SAML v2.0 est actuellement le protocole le plus utilisé pour les échanges des identités, de l'authentification, des attributs et les informations d'autorisation entre les domaines de sécurité. Ainsi, nous avons principalement à introduire dans l'architecture IMS deux entités SAML spécifiques, l'IdP (fournisseur d'identité) et les différents SP (fournisseurs de services). L'IdP va authentifier l'utilisateur grâce à son identité IMS, émettre des affirmations d'identité SAML, tandis que le SP recevra et validera les affirmations.
L'IdP peut être fusionné avec une entité IMS, mais il nécessite de modifier les implémentations actuelles du cœur IMS. Il est possible d'éviter cela en ajoutant IdP et SP en tant que nouvelles entités SIP avec des capacités SAML améliorées. Chaque fournisseur de services doit posséder un proxy SIP comme étant SAML SP capable d'interpréter des messages SAML relié à ses différents AS.

La figue 5.11 présente essentiellement les différentes entités du SSO dans une architecture IMS pour l'accès aux services des tiers. Entre le noyau IMS et le réseau du fournisseur de service se trouve un proxy SIP agissant comme un IdP SAML qui sera en mesure de transmettre des messages SIP vers le réseau du fournisseur de services contenant des affirmations d'identité SAML. Cet IdP est connecté à une base de données d'identité permettant de créer les assertions d'identité, notamment par l'enregistrement des fédérations d'identité. Entre l'IdP et les serveurs d'applications, se trouve dans le réseau du fournisseur de

services un proxy SIP fonctionnant comme un SAML SP. Le SAML SP aura comme fonctions la demande et la vérification des affirmations d'identité SAML ainsi que l'exécution d'autorisation de l'utilisateur grâce à la connexion à un serveur de profils contenant la correspondance entre les pseudonymes et les identités locales et l'information des utilisateurs spécifiques au service.

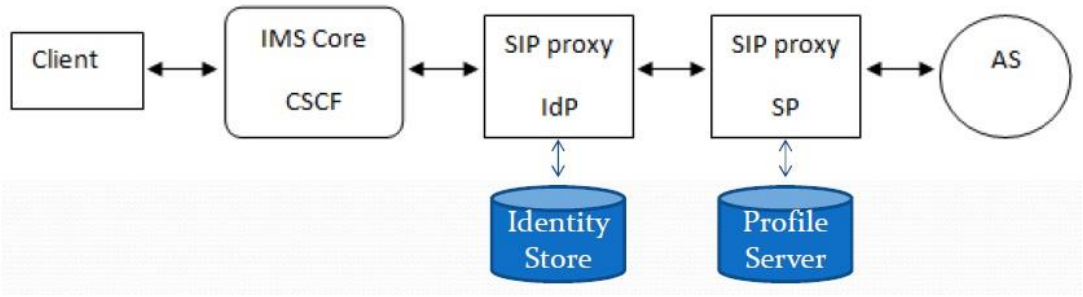


Figure 5.11. Architecture de base SSO pour IMS

Pour l'ajout du l'IdP il existe principalement deux possibilités, l'IdP peut être soit situé à l'intérieur du réseau IMS de l'utilisateur ou à l'extérieur de ce réseau en tant que fournisseur d'identité tiers.

Avec un fournisseur d'identité tirs l'IdP ne se trouve pas dans le domaine de l'opérateur IMS. Cela implique que l'IdP doit utiliser un mécanisme d'authentification pour identifier l'émetteur de la requête SIP. Cependant, puisque l'utilisateur est déjà authentifié par le réseau IMS via le protocole IMS-AKA, on peut utiliser ce contexte d'authentification pour permettre une authentification directe de l'UE avec l'IdP. En effet, le modèle 3GPP GBA [Gpp12, ShMa06], qui fait partie de l'architecture d'authentification générique (*Generic Authentication Architecture GAA*), définit une procédure de bootstrapping basée sur AKA pour authentifier un utilisateur à une application [Gpp05]. De cette façon, nous utilisons le mécanisme d'authentification IMS-AKA et ensuite avons un coût de mise en œuvre faible, ainsi qu'un mécanisme de SSO totalement transparente pour l'utilisateur.

La deuxième façon de fournir SSO est de placer l'IdP à l'intérieur du réseau IMS. En effet à la procédure d'enregistrement du client au réseau IMS, l'UE est authentifiée par IMS-AKA. Après cette phase, en établissant un lien sécurisé (tunnel IPsec) avec l'UE, le P-CSCF est le garant de l'identité du client à l'intérieur du réseau IMS.

Comme expliqué dans la section 5.1.3, les en-têtes SIP *P-Preferred-Identity* et *P-Asserted-Identity* sont utilisées dans le réseau IMS pour assurer l'authentification des requêtes SIP après l'enregistrement de l'utilisateur.

Puisque dans cette architecture l'IdP est situé à l'intérieur du réseau IMS de confiance, les requêtes SIP qui seront reçus par l'IdP contiendront l'en-tête P-Asserted-Identity avec l'identité publique IMPU du client. Il n'est pas donc nécessaire pour l'IdP d'utiliser un mécanisme d'authentification à nouveau, ce qui évite une nouvelle étape d'authentification qui serait consommatrice de ressources et de temps.

Cette architecture SSO est alors essentiellement basée sur IMS-AKA. Cependant cela ne peut fonctionner que si l'IdP est situé à l'intérieur du réseau de l'opérateur IMS. Si ce n'est pas le cas, un autre mécanisme d'authentification devrait être utilisé, comme vu précédemment.

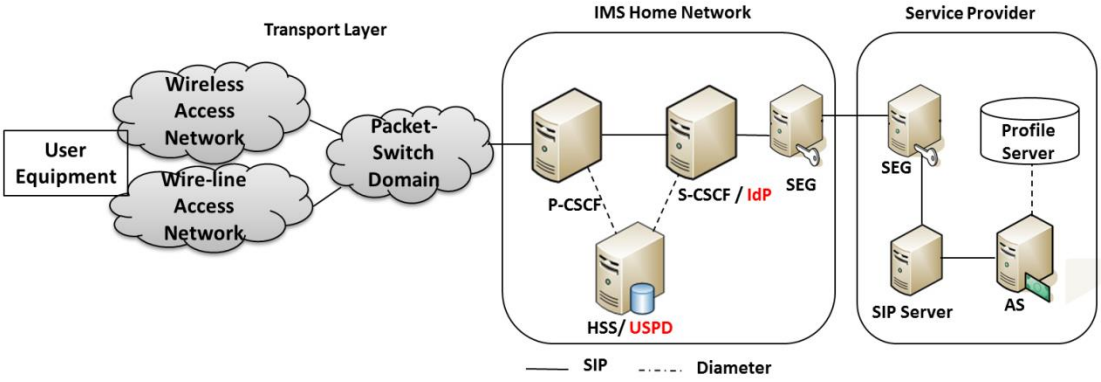


Figure 5.12. L'architecture IMS avec l'IdP dans le réseau IMS

Par conséquent, la meilleure façon de fournir SSO est de placer l'IdP à l'intérieur du réseau IMS. La figure 5.12 présente plus de détails comment intégrer les composantes SSO au niveau d'une architecture IMS avec l'IdP à l'intérieur du réseau IMS. En effet, l'IdP est fusionné avec le S-CSCF. Dans le réseau du fournisseur de services, le serveur SIP agit comme SAML SP. Il est relié à l'AS par l'intermédiaire de l'interface ISC et au serveur de profils, qui contient les informations de profil de l'utilisateur et permet d'assurer l'autorisation. En outre il y a une base de données USPD (User Subscription Profil Database) ajoutée au HSS. L'USPD conserve les informations de profil de l'utilisateur en ce qui concerne son abonnement à des tiers fournisseurs de services tiers [IsGr11].

5.4.3. Etablissement de la fédération

5.4.3.1. Introduction à la confiance

La fédération consiste à définir une architecture permettant le transport d'informations portant sur des identités numériques. Ceci, dans le but d'offrir l'accès à des ressources de domaines de sécurité différents de ceux d'appartenance des identités. Ce principe gravite autour de trois problématiques.

5.4.3.2. La délégation des tâches administratives

Dans un environnement où chaque application est en charge d'assurer l'exécution des procédés administratifs de gestion des identités, la saisie répétée d'identifiants est un problème majeur en termes de confort d'utilisation des applications et de transmission répétée d'identifiants sur le réseau. Les architectures d'authentification unique basées sur la délégation des procédés administratifs auprès de tiers de confiance apportent une solution à ce problème. Ce type d'architectures repose sur l'établissement de liens de confiance entre fournisseurs de services et autorités en charge des procédés administratifs. Les entités ainsi liées forment un cercle de confiance. Les fournisseurs de services acceptent les affirmations faites par les autorités administratives, qui ont la charge de fournir la preuve de leur identité auprès des fournisseurs de services. Cela se traduit dans la pratique par le partage de secrets permettant la signature des affirmations, généralement, grâce à une infrastructure à clés publiques.

5.4.3.3. L'ouverture des systèmes d'informations

L'ouverture des systèmes d'informations permet la mise à disposition de ressources internes à des intervenants extérieurs, humains ou logiciels. Pour des raisons évidentes de charge administrative, et donc de coût financier, un même système d'informations ne peut directement prendre en charge la gestion des identités des systèmes tiers auxquels appartiennent ces intervenants. Cette ouverture doit donc s'accompagner de l'ouverture des systèmes de gestion des identités, et de la mise en place d'infrastructures d'interconnexion de ceux-ci. L'établissement de liens de confiance entre les systèmes d'informations souhaitant interopérer est donc dans ce cas également justifié. Les systèmes d'informations appartenant globalement à des entités organisationnelles tierces, la notion de fédération prend tout son sens. Il s'agit, dans un premier temps, de formaliser administrativement un partenariat afin de former une fédération. Cette fédération définit ensuite sa propre politique de gestion des identités, entre systèmes, et basée sur des liens de confiance.

5.4.3.4. Sécurisation des architectures orientées services

L'interconnexion d'applications intra ou inter-systèmes d'informations, et la conception d'applications inter-systèmes d'informations, sont des problèmes complexes en terme de gestion des identités. Il est en effet nécessaire d'assurer la sécurité des échanges, mais aussi de gérer les identités de clients applicatifs autonomes. Qui plus est, une architecture de fédération d'identités basée sur les technologies du Web est une architecture orientée services.

Elle apporte une couche de sécurité et de gestion des identités au sein d'une architecture orientée service existante, en conditionnant les échanges de messages, et en ajoutant des informations de sécurité aux entêtes des messages échangés.

5.4.3.5. Identités locales d'utilisateurs

Dans un modèle SSO fédéré, l'utilisateur doit avoir des identités locales à la fois au niveau de l'IdP et le SP, et une fédération associe ces deux identités locales en utilisant un pseudonyme. Il est intéressant de noter que l'identité locale ne sera jamais reconnue ou utilisé en dehors de son domaine local.

Un utilisateur dispose d'une identité locale IM (IMLI) à chaque réseau de SP, qui est créé lors de la souscription de l'utilisateur au SP. Au cours de la procédure d'installation de la fédération (par une méthode dynamique reliant les comptes [Hloc08]ou par n'importe quelle méthode hors ligne), l'IdP crée un pseudonyme pour chaque SP, et associe l'identité locale avec le SP à travers ce pseudonyme.

Par conséquent, le pseudonyme est la seule identité reconnue à la fois par l'IdP et le SP. Les bases de données d'identités tenues par l'IdP et le serveur SIP, détient le tuple de l'identité locale, le pseudonyme et l'adresse de l'autre domaine (par exemple l'IdP ou le SP).

5.4.3.6. Procédure de construction de la fédération

Puisque un utilisateur peut avoir des identités différentes dans différents domaines, le but de la solution SSO est d'établir et utiliser une fédération des identités différentes liées à un utilisateur particulier. Il doit y avoir un accord entre les fournisseurs sur un ensemble d'identifiants et/ou des attributs d'identité par laquelle les sites seront référer à l'utilisateur.

Cet accord devrait porter sur un certain nombre de questions, telles que l'existence d'identités locales d'un même utilisateur dans chaque domaine, la façon d'établir la fédération, dynamique ou basée sur des identités fédérées préétablies, la persistance des identificateurs fédérés, ou l'échange des attributs de l'utilisateur. Avant SAML v2.0, les versions précédentes autorisés à effectuer un accord hors bande (*out of bound*), sur les types d'identifiants qui pourraient être utilisés pour représenter une identité fédérée entre les partenaires. L'établissement de la fédération n'était pas possible en échangeant des messages SAML. Depuis l'introduction de SAML v2.0, il est possible, en échangeant des messages SAML d'établir dynamiquement une fédération d'identité, ainsi que de préserver l'anonymat de l'utilisateur en utilisant des alias de fédération.

Une assertion SAML comprend un identifiant unique appelé *nameID*. Il peut directement identifier un utilisateur ou bien il peut être utilisé comme un pseudonyme. L'utilisation d'un pseudonyme peut être utile pour protéger la vie privée et assurer l'anonymat de l'utilisateur. La spécification SAML définit deux alias : l'identifiant provisoire et l'identifiant persistant. L'identifiant provisoire est un alias temporaire qui peut changer après chaque session. Le sujet ne peut donc pas être associé à un compte local du fournisseur de service. Cette méthode d'alias semble être bien adaptée à l'approche coopérative, mais ne convient pas bien notre objectif de maintenir des comptes locaux à chaque fournisseur de services. L'identifiant persistant (pseudonyme invariant) est donc plus adaptée à nos besoins car il ne change pas au fil du temps et permet de lier des comptes d'utilisateurs en préservant l'anonymat.

La figure 5.13 présente un cas d'utilisation de fédération d'identité avec pseudonyme persistante dans le contexte de Web-SSO, qui doit être adapté à notre architecture en utilisant les liens appropriés.

Avec l'architecture proposée, il est possible d'utiliser soit une fédération hors-bande ou une fédération dynamique en utilisant des identificateurs persistants pour créer un alias et effectuer la cartographie des IMPU avec IMLI (IM identité locale) l'identité connue par les fournisseurs de services [IsGr09, Jean09].

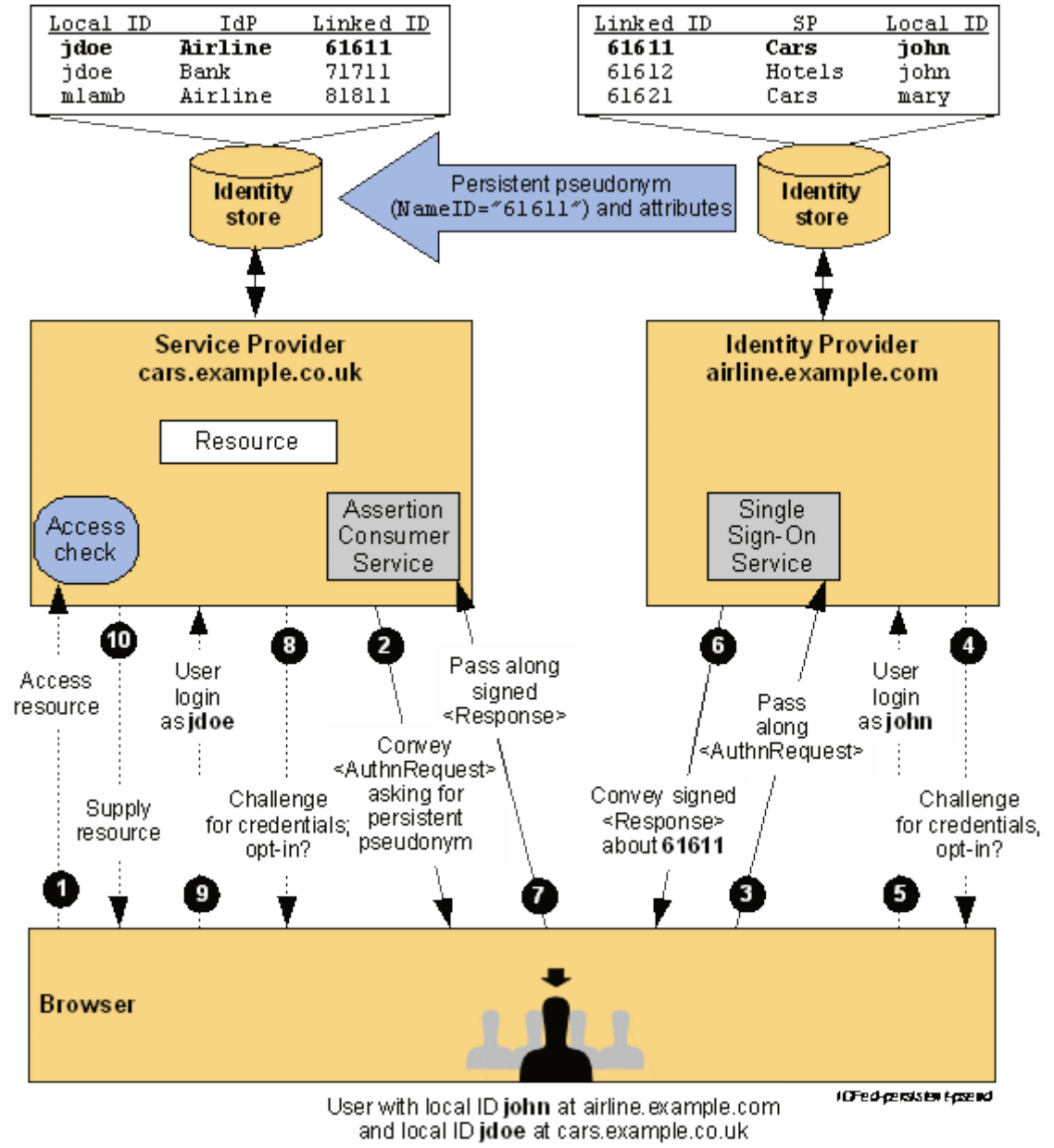


Figure 5.13. Fédération d'identité avec Pseudonyme persistante

5.4.3.7. Protection de la vie privée

Les échanges d'attributs associés à un utilisateur entre l'IdP et le SP font appel à des pseudonymes connus des deux entités (appelés alias), via un canal externe. La norme précise aussi si l'utilisateur donne son consentement pour une authentification SSO seule, ou pour le partage d'attributs. Un utilisateur peut retirer son consentement de fédération à un IdP ou SP en pratiquant la défédération. C'est-à-dire, il peut faire une défédération auprès d'un IdP et ainsi empêcher un SP de recouper ses actions préalables avec les suivantes. Comme l'usage simultané de pseudonymes différents avec un SP n'est pas possible, la défédération offre à l'utilisateur la possibilité d'interagir avec un SP sous deux pseudonymes différents à deux moments différents [Kbek10].

Reste comme même un autre problème, c'est celui du traçage possible des activités de l'utilisateur par l'IdP, cependant une solution a été apporté avec les nouvelles spécifications Liberty, où les attributs de l'utilisateur sont stockés localement par ce dernier.

Dans le cas où l'utilisateur est une entreprise ou une organisation cette solution est possible et même préférable, mais dans le cas où l'utilisateur est un simple abonné cette solution devient un peu difficile vu que c'est l'utilisateur qui doit gérer les différents *mapping* seul. Peut-être avec la vitesse actuelle du développement des applications pour les smartphone cela peut-être possible d'ici peu. Pour le moment on est obligé de faire confiance à l'opérateur pour qu'il respecte le contrat signé avec ses abonnés pour assurer la protection de leurs vies privées selon les normes et lois qui le gèrent.

5.4.4. SIP SAML : Profil et binding

Nous avons décidé d'utiliser SAML pour la fédération et la propagation d'identité dans l'architecture SSO d'IMS. Cela nécessite que l'environnement IMS supporte le protocole SAML par le biais de profils et binding adéquats. En effet, la correspondance des échanges des messages demande-réponse SAML avec des protocoles de communication standards, appelées binding SAML, a besoin d'être spécifiée et décrite avec suffisamment de détails pour s'assurer de l'interopérabilité.

OASIS définit par exemple le binding SAML SOAP décrivant comment SAML et les échanges des requêtes et réponses devraient être mis en correspondance avec les échanges de messages SOAP. Etant donné que le réseau IMS est basé sur SIP, il est nécessaire d'utiliser un binding adéquat compatible avec le protocole SIP. Cependant, jusqu'à ce jours aucun binding SAML pour SIP n'est standardisé ou un système équivalent existe. En effet, tous les binding et les profils spécifiés par OASIS s'appuient sur l'utilisation du protocole HTTP. Quelques suggestions ont néanmoins été soumises, dans le projet Liberty Alliance [Libe06], ou comme un *draft* IETF [Htsc10].

5.4.4.1. SAML basé sur HTTP-URI pour un binding avec le protocole SIP

Le draft IETF [PPSH10] spécifie un profil complet en utilisant SAML basé sur HTTP-URI pour un binding avec le protocole SIP. Ce profil est appelé "*AS-driven*", car ce n'est pas le SP qui initie l'échange des requêtes/réponses SAML. Au lieu de cela, un SIP INVITE (ou

une autre méthode SIP) sera envoyé à l'AS et l'IdP qui est situé sur le chemin va initier l'échange SAML en attachant l'URI de l'assertion qui sera récupérée par les SP.

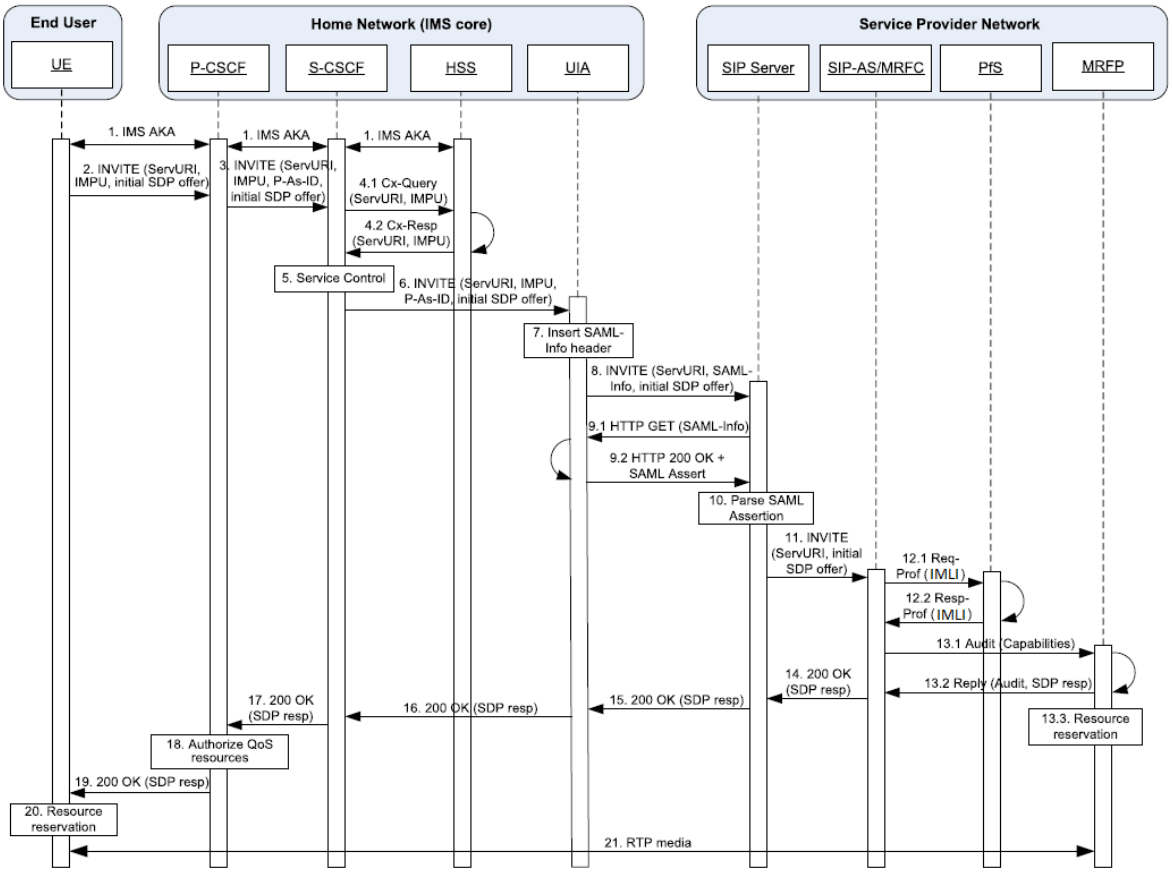


Figure 5.14. Séquence SSO initiée par l'IdP pour des applications IMS (idP à l'intérieur du réseau IMS) [IsGr11]

La figure 5.14 présente le flux de messages par rapport à ce profil dans notre contexte IMS.
1. L'utilisateur s'authentifie auprès du réseau IMS avec le protocole IMS-AKA.
2-3. L'utilisateur effectue une demande SIP INVITE à un serveur d'application tiers.
4-5. Le S-CSCF vérifie l'autorisation de l'utilisateur pour accéder au serveur d'application demandé.
6. Le S-CSCF transmet la demande à l'IdP.
7-8. L'IdP anonymise le message SIP INVITE, en effaçant le champ "Contact" du message et en remplaçant le champ "From" avec une identité anonyme. L'IdP ajoute l'URI HTTP de l'assertion SAML, nommé "SAML info-header" et transmet le message aux fournisseurs de service.

9-10. Le serveur SIP dans les réseaux des fournisseurs de services SIP reçoit le message INVITE anonyme et utilise "l'info-header" pour obtenir l'affirmation de l'identité SAML correspondant à l'émetteur de la demande avec le protocole HTTP. Il vérifie l'assertion SAML reçu et récupère l'identité d'utilisateur local (IMLI) grâce au pseudonyme. 11-12. Le serveur SIP transmet ensuite le message SIP INVITE à l'AS qui sera en mesure de récupérer les informations de profil de l'utilisateur grâce à l'IMLI. 13-20. Le reste de la séquence est la même que pour n'importe quel établissement de session IMS.

Etant donné que dans le système SSO proposé, la fédération d'identité est réalisée avec des pseudonymes persistants et il n'est pas nécessaire d'échanger des attributs puisque l'information de l'utilisateur est directement situé sur chaque fournisseur de services utilisateur, la seule information pertinente qui doivent être portés par l'assertion SAML est le pseudonyme de l'émetteur du message SIP, comme illustré sur la figure 5.15.

```
<Assertion   ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
             IssueInstant="2012-03-13T00:46:02Z" Version="2.0"
             xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
      <Issuer>
             IMSoperator.com
      </Issuer>
             <Subject>
                    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
                           UB/WJAaKAPrSHbqlbcKWu7JktcKY
                    </NameID>
             </Subject>
             <Conditions NotBefore="2003-04-17T00:46:02Z" NotOnOrAfter="2003-04-17T00:51:02Z">
                    <AudienceRestriction>
                           <Audience>
                                  ServiceProvider.com
                           </Audience>
                    </AudienceRestriction>
             </Conditions>
             <AuthnStatement   AuthnInstant="2012-03-17T00:46:02Z"
                               SessionIndex="_a75adf55-01d7-40cc-929f-dbd8372ebdfc">
                    <AuthnContext>
                           <AuthnContextClassRef>
                                  urn:oasis:names:tc:SAML:2.0:ac:classes:Password
                           </AuthnContextClassRef>
                    </AuthnContext>
             </AuthnStatement>
</Assertion>
```

Figure 5.15. Assertion SAML

5.4.4.2. SAML basé sur SIP-URI en utilisant des messages SOAP sur SIP

La principale critique que l'on pourrait formuler contre le profil du binding précédent est l'utilisation d'échanges de messages http, ce qui implique que les proxys SIP doivent

également être équipés d'une pile de protocole HTTP. Nativement, un cœur de réseau IMS prend uniquement en charge les protocoles SIP et Diamètre. Par conséquent, un moyen de transmettre les demandes/ réponses SAML à l'intérieur des messages SIP serait plus approprié. C'est la raison pour laquelle le binding SAML basé sur des URI SIP (*SAML SIP-URI-based Binding : SSUB*), en association avec le binding SAML SOAP sur SIP (*SAML SOAP Binding over SIP : SSSB*) a été proposé dans [NTTH09].

Comme le binding SAML basé sur des URI HTTP pour le protocole SIP, *SSUB* étend le binding SAML URI s'appuyant sur le fait que la spécification de ce binding est indépendante du protocole de transport sous-jacent du processus de résolution d'URI, et utilise donc des URI SIP. Ensuite, le protocole SIP sera utilisé pour effectuer la résolution URI, ce qui impliquera un message SOAP échangé sur SIP, d'où le SAML liaison SOAP sur SIP (SSSB). SSSB est basée sur le binding SOAP SAML [Cant05] et l'utilisation du protocole SIP pour transporter des messages SOAP. Ce binding a été proposé dans un draft IETF [Ndea00] par l'ajout d'une nouvelle méthode SIP appelé SERVICE.

L'avantage de ces bindings est le fait qu'ils ne nécessitent plus une implémentation de la pile HTTP dans les proxys SIP, mais ils nécessitent de modifier un peu le protocole SIP et d'ajouter la méthode SERVICE.

5.4.5. SSO Multi-niveaux

Le SSO permet de diminuer les contraintes et de faciliter la vie de l'utilisateur, en plus le SSO renforce la sécurité lorsqu'un dispositif d'authentification forte remplace une multitude de couples identifiant-mot de passe. Par contre le SSO peut aussi affaiblir la sécurité en ce qu'une usurpation d'identité réussie par un pirate lui donne accès à l'ensemble des applications concernées. En effet, une seule authentification valide suffit pour que l'utilisateur soit ensuite automatiquement authentifié auprès des autres applications, qui peuvent très sensibles comme les applications bancaires, à savoir Payement en ligne, facture électronique et enchères en ligne.

Une façon d'améliorer la solution de basique SSO est d'utiliser plusieurs mécanismes d'authentification, chacun de ces mécanismes est associé à un certain nombre d'applications selon un niveau d'assurance (LOA), de cette façon on a pour chaque paquet de service une architecture SSO basée sur un mécanisme d'authentification.

5.4.5.1. Amélioration d'IdP

Pour activer l'architecture multi-niveaux de nouvelles fonctionnalités doivent être ajoutées à certaines entités, principalement l'IdP. En effet, l'IdP doit prendre en considération les différents niveaux d'assurance (LoA) et doit être en mesure d'authentifier l'utilisateur via différentes stratégies d'authentification. La façon dont il se comportera est similaire au système SHARE proposé dans [YiYH09]. L'IdP maintient aussi une correspondance des SPs avec leur niveau respectif, enregistré lors de l'établissement de la fédération, ainsi que le niveau actuel de l'utilisateur. Il doit également inclure des informations sur le niveau LoA de l'utilisateur dans l'assertion SAML. En outre, il doit adopter un comportement correct quand un client essaie d'accéder aux différents niveaux de serveurs d'applications.

5.4.5.2. Les stratégies d'authentification LOA

Comme l'IdP se situe entre le client et le SP dans le chemin des messages SIP, il va vérifier le niveau actuel d'authentification de l'utilisateur après avoir reçu un message d'invite SIP et avant de le transmettre au SP. Si le niveau du SP est supérieur au niveau actuel de l'utilisateur, l'IdP va demander à l'utilisateur de s'authentifier à l'aide de la stratégie d'authentification du niveau de la SP invité, par l'envoi d'une réponse non autorisé SIP 401 spécifiant le type d'authentification prévu. Ensuite, l'utilisateur envoie de nouveau le message SIP INVITE avec les bonnes crédentiels. Après vérification par l'IdP, le message SIP sera transmis normalement avec en plus l'URI de l'affirmation (insérée par l'IdP). Le flux de messages associé à ce comportement est illustré sur la Figure 5.16.

Le SP peut aussi vouloir connaître le contexte de l'authentification liée à une assertion bien précise. SAML permet d'ajouter dans les assertions le niveau LoA correspondant. OASIS est entrain de définir un processus de définition des schémas pour échanger des informations LoA dans les assertions SAML [Oasi09].

Dans la figure 5.16, nous donnons un exemple avec deux LoA utilisant IMS-AKA pour le niveau 1 (le niveau le plus bas) et une autre stratégie d'authentification pour le niveau 2. En fait, dès que le client accède au réseau IMS, l'utilisateur est authentifié auprès de l'IdP grâce à IMS-AKA. Ainsi IMS-AKA sera toujours le mécanisme d'authentification correspondant au niveau le plus bas de l'assurance. Cependant IMS-AKA avec UICC (Universal Integrated Circuit Card) est déjà un solide mécanisme à deux facteurs d'authentification basée sur "quelque chose que vous savez", le code PIN, et "quelque chose que vous avez" l'UICC.

Selon le guide d'authentification électronique publié par NIST [BuDP06], qui classe les mécanismes d'authentification sur 4 niveaux, sur la base de l'UICC IMS-AKA peut être considéré au niveau 3. En outre, tous les autres mécanismes d'authentification mis en œuvre dans un environnement SIP sont moins sûrs que l'IMS-AKA.

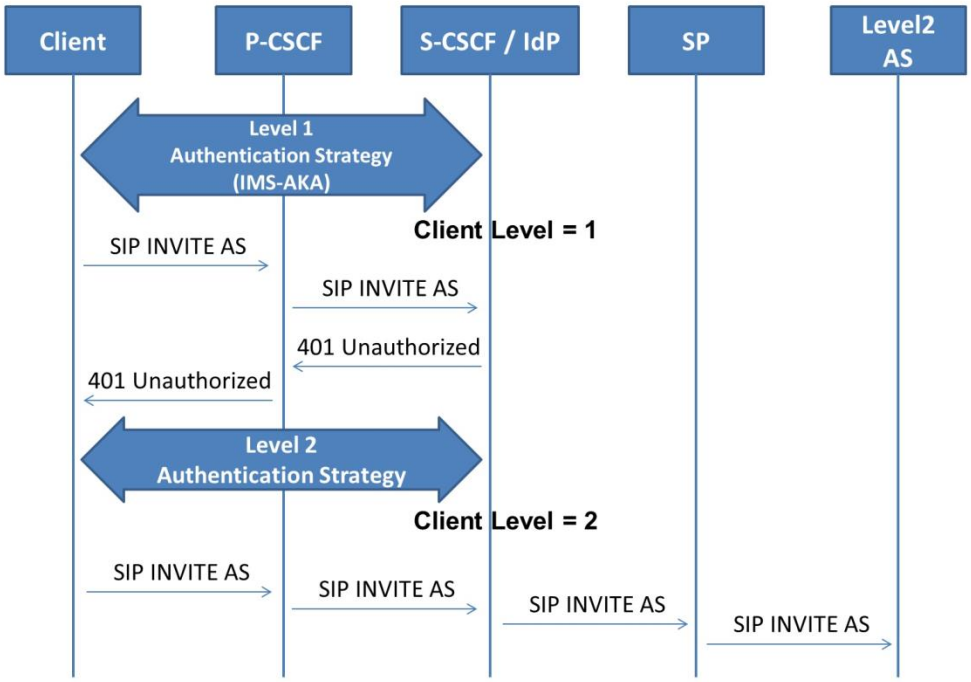


Figure 5.16. Exemple multi-niveau avec deux niveaux LoA

Pour garder la délégation de l'affirmation de l'identité par P-CSCF, avec l'en-tête P-Asserted-Identity, après avoir effectué l'authentification, un nouveau niveau d'authentification doit également utiliser un mécanisme d'accord clé d'authentification et permettant de renouveler les clés IPSec du tunnel établi entre le client et le P-CSCF lors IMS-AKA.

Il est à noter que notre modèle est indépendant du mécanisme d'authentification utilisé, et il peut être étendu à plus de deux niveaux en fonction des besoins et de la classification des services d'application.

5.4.5.3. Exemple d'implémentation d'un système avec deux niveaux LoA :

Pour être plus sécurisé que l'IMS-AKA, le nouveau mécanisme d'authentification peut utiliser la cryptographie asymétrique au lieu d'employer des clés secrètes partagées à long

terme. Ce nouveau mécanisme se base sur l'échange Diffie-Hellman pour assurer l'authentification et un nouvel accord sur les clés de session.

La figure 5.17 ci-dessous illustre la mise en place du nouveau tunnel IPSec entre le client et le P-CSCF, ainsi que l'authentification du client par l'IdP, en utilisant l'échange de clés Diffie-Hellman combiné avec les signatures numériques.

L'utilisation des certificats pour les utilisateurs pose cependant le problème concernant leur stockage et distribution. Nous suggérons que les certificats soient stockés sur un périphérique physique, autre que l'UICC, comme une carte mémoire ou un périphérique USB. De cette manière on ne va pas affecter les deux techniques d'authentification si l'UICC est compromise. Le tableau 5.1 ci-dessous résume les deux LoA définis :

Tableau 5.1. Exemple d'un système d'authentification avec deux niveaux LoA

Niveau	1	2
Mécanisme d'authentification et d'accord de clés	IMS-AKA	Authentification avec Diffie-Hellman utilisant la signature numérique

5.5. Conclusion

Dans ce chapitre nous avons présenté un nouveau modèle, nommé *virtual walled-garden*, de fourniture de services centré sur l'utilisateur en IMS. Ce modèle de fourniture de service permet d'offrir plus de liberté d'utiliser les services de tout fournisseur de contenu en fonction des besoins et préférences des utilisateurs. De cette manière les trois parties (utilisateur, fournisseurs de services et opérateur IMS) sont satisfaites. Les utilisateurs auront accès à un plus large éventail de services soutenus par l'IMS, les fournisseurs de services peuvent mettre en œuvre un large éventail de services IMS/SIP sans aucun investissement sur la mise en œuvre d'un réseau de cœur IMS ou de sa maintenance. Quant aux opérateurs cette façon de faire constitue une nouvelle forme de partenariat d'affaires avec les fournisseurs de services. Le modèle *virtual walled-garden* se base sur une fédération d'identité multi niveaux pour prendre en considération plusieurs niveaux de sécurité selon la criticité des applications sollicitées.

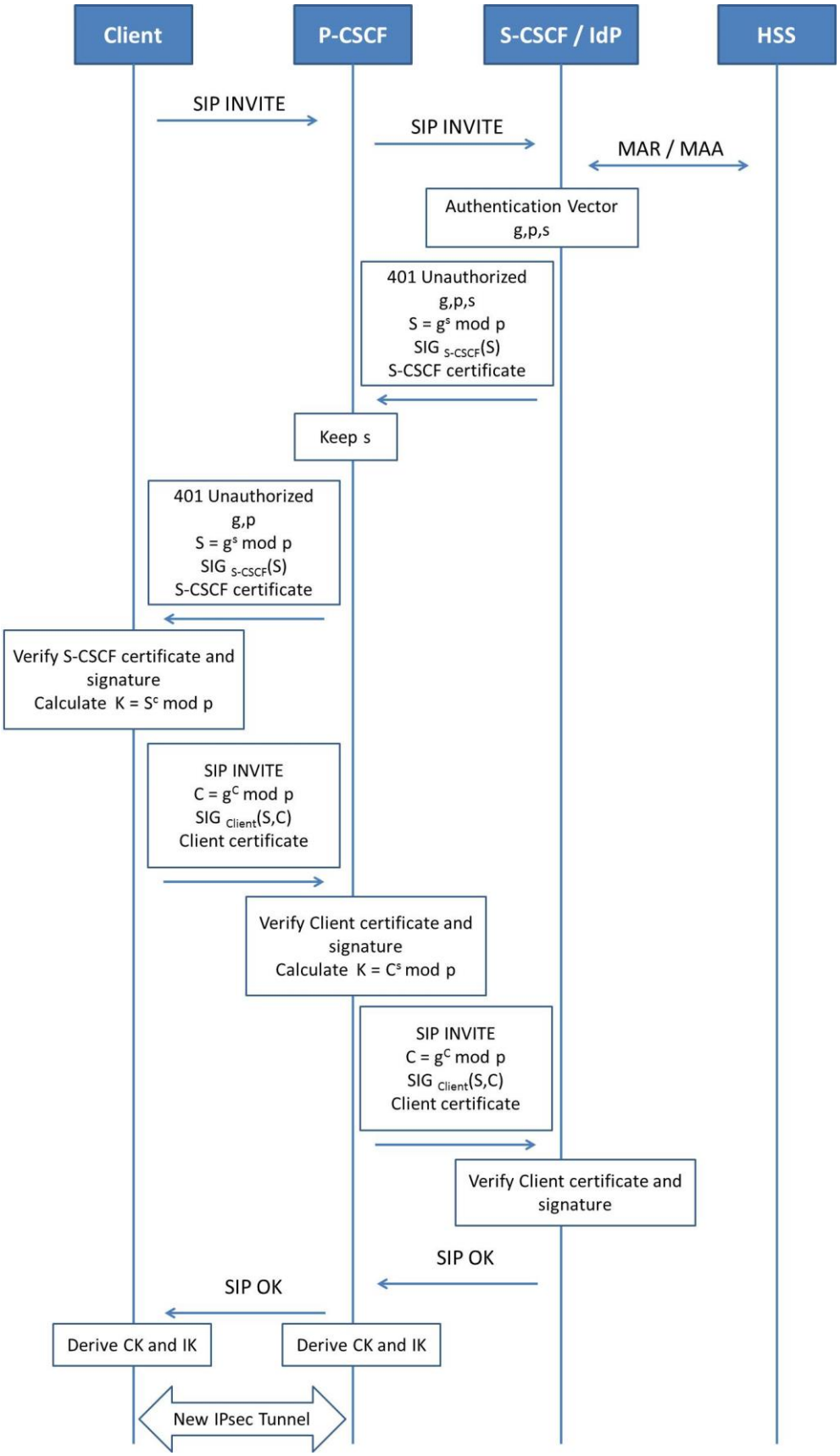


Figure 5.17. Authentification avec Diffie-Hellman utilisant la signature numérique pour IMS

Conclusions et perspectives

Conclusions

Les réseaux de nouvelles générations, connus sous l'acronyme NGN ou IMS, font référence à une architecture réseau en couche découplant les fonctions de contrôle, de transport et de fourniture de service. En effet, l'architecture IMS repose sur la distribution de l'intelligence à travers les différentes composants du réseau afin d'assurer une meilleure résistance au facteur d'échelle couplée à une indépendance inter-couche. La dite indépendance permet d'envisager des architectures multi-intervenants. Toutefois, cet aspect ouvert de l'architecture NGN présente un certain nombre de défis à relever notamment en termes de sécurité. Pour relever ce défi la plupart des opérateurs ont opté pour une architecture fermée en utilisant le modèle walled-garden. Certes avec ce type de déploiement les opérateurs créent un univers autonome dans lequel les abonnés sont autorisés à profiter de tous les services et les contenus offerts seulement par leur opérateur, dans un environnement entièrement sécurisé. De cette façon, les opérateurs ont l'assurance que les abonnés n'auront jamais avoir à chercher des contenus ou des services à l'extérieur du réseau, et, par conséquent, tous les revenus générés vont directement à l'opérateur. Malheureusement, ce modèle limite les utilisateurs finaux seulement aux services que leurs opérateurs IMS offrent. Par conséquent, le modèle walled-garden ne parvient pas à créer une demande massive des utilisateurs, qui est la principale force motrice du chiffre d'affaire.

Afin de donner une certaine liberté aux utilisateurs d'utiliser les services de tout fournisseur de contenu l'ouverture des opérateurs IMS auprès de tiers fournisseurs de services tiers tend à être une obligation. D'autre part, les opérateurs préfèrent garder le contrôle. Il faut donc trouver une solution entre les deux qui permet d'instaurer un environnement de confiance entre les trois parties : utilisateur, opérateur et partie tierce, c'est dans ce contexte que se situent les travaux de la thèse.

Notre première solution "*The PKI-based One-way IMS-AKA*" propose un nouveau protocole d'authentification et d'accord de clé (*Authentication and key agreement*) au niveau IMS plus sécurisé. Ce protocole basé sur une infrastructure PKI permet de résoudre les problèmes de sécurité dont l'IMS-AKA souffre, conserve le one-way et améliore les échanges en termes de

performance (par rapport à l'IMS-AKA). Sur la base des résultats obtenues, nous pouvons confirmer que notre protocole (1) peut économiser au moins 21,5% (pire cas) du trafic SIP/Cx par rapport à l'IMS-AKA, (2) permet de réduire la consommation de la bande passante de 27% par rapport à l'IMS-AKA, (3) résiste aux attaques atteignant la confidentialité et l'intégrité des données lors d'un enregistrement IMS (validé par AVISPA), et (4) réduit le temps de réponses des requêtes d'enregistrement IMS.

Le modèle virtual walled-garden a été notre deuxième contribution pour la fourniture de services centré sur l'utilisateur en IMS. Ce modèle de fourniture de service permet d'offrir plus de liberté d'utiliser les services de tout fournisseur de contenu en fonction des besoins et préférences des utilisateurs. De cette manière les trois parties (utilisateur, fournisseurs de services et opérateur IMS) sont satisfaites. Le modèle virtual walled-garden permet de créer un lien de confiance entre le domaine IMS et les services externes, et permet de réduire la charge des utilisateurs ainsi que les SPs par l'utilisation d'un mécanisme SSO en utilisant une fédération d'identité multi niveaux pour prendre en considération plusieurs niveaux de sécurité selon la criticité des applications sollicitées.

Perspectives

En perspective de notre étude, nous envisageons l'extension de nos travaux par l'étude des points suivants :

Un seul protocole d'authentification IMS

Le protocole d'authentification proposé (One-way) permet de gagner en performance pour les utilisateurs utilisant des réseaux d'accès sans-fil, et avec comme hypothèse l'utilisation d'une carte ISIM/USIM. Une question s'impose : si on accède à l'IMS via un réseau fixe, en utilisant un terminal qui ne possède pas de carte ISIM/USIM comment authentifier les utilisateurs tout en gardant une bonne performance et un niveau de sécurité acceptable ?

Pour répondre à cette question nous pouvons étendre notre protocole. Pour cela nous pouvons se baser sur les mécanismes de la virtualisation en utilisant des clients légers/virtuels par exemple, nous pouvons aussi utiliser l'architecture GAA.

Les profils utilisateur et la vie privée

Le maintien de profils d'utilisateurs de manière évolutive tout en assurant le respect de la vie privée de l'utilisateur est un grand défi pour un modèle IMS centrée sur l'utilisateur. Dans un modèle fermé (walled-garden), au niveau IMS, les profils utilisateur sont maintenus par le HSS, un référentiel central pour le cœur du réseau IMS. Les serveurs d'application (AS), qui sont sous le contrôle de l'opérateur de base IMS, reçoivent des informations de l'utilisateur requises à partir du HSS via une interface intra-opérateur sécurisé (Sh). Toutefois, dans le cas d'un modèle IMS avec des tiers, par exemple notre modèle le virtual walled-garden le nombre de places où les profils utilisateur vont être maintenus va croître chaque fois que l'utilisateur s'abonne à un service d'un nouveau SP, un profil d'utilisateur est créé au niveau du PfS (Profil Server) de ce SP. Dans un souci d'assurer la vie privée de l'utilisateur, il faut réduire au minimum le volume des informations conservées par le PfS, le HSS aussi ne doit fournir que les informations spécifiques à l'utilisateur.

SSO pour plusieurs appareils physiques en même temps

Les systèmes SSO (Single Sign-On) traditionnels fonctionnent bien lorsqu'il est question d'utiliser un seul poste de travail et d'accéder aux services classiques Client/Serveur. Le problème se pose lorsque l'utilisateur a besoin d'utiliser plusieurs appareils physiques en même temps.

Habituellement, les utilisateurs possèdent plusieurs appareils pour effectuer des tâches différentes sur chacun d'eux. Par conséquent, les utilisateurs doivent se connecter et s'authentifier sur chaque périphérique. Les systèmes traditionnelles de SSO ne fonctionnent pas bien dans ce cas, car l'utilisateur doit entrer au moins un mot de passe par machine. Par conséquent, il n'y a pas de single sign-on. Pour bien comprendre l'idée voici un scénario explicatif :

Scénario explicatif :

Bob est propriétaire de trois terminaux: un smartphone, un ordinateur portable et un poste de travail. Lorsque Bob rentre à sa maison (Home Network), il s'authentifie auprès de son smartphone (le master). En ce moment même, "notre agent" (qu'il faudra développer) se lance dans le smartphone et lit un fichier depuis une carte SD insérée dans l'appareil ou bien depuis un serveur au sein du réseau d'opérateur. En utilisant le mot de passe entré auparavant, l'agent

déchiffre le fichier. Ensuite, il lit tous les secrets (Credentials) appartenant à Bob (mots de passe, clés, etc.) et les stocke dans la mémoire du processus de l'agent (ce qui n'est pas accessible à partir d'autres procédés ou machines).

Lorsque Bob démarre son ordinateur portable, un agent client est exécuté. Ensuite, l'ordinateur portable demande à son agent local de s'authentifier. L'agent de l'ordinateur portable découvre que le master appartenant à l'utilisateur Bob est en cours d'exécution (le smartphone) et se rend compte que cet agent a tous les secrets de Bob. Au lieu de demander à Bob de s'authentifier, l'agent de l'ordinateur portable demande à l'agent du smartphone tous les secrets nécessaires. Et le même processus se passe pour la station de travail.

Il faut noter que chaque fois qu'il y a une requête depuis une machine de Bob vers le master, le smartphone demande la confirmation. Par la suite, il suffit que Bob confirme ou rejette la demande.

Liste des publications

- **Conférences et workshops internationaux :**
  - Mohamed Maachaoui, Anas Abou El Kalam, Christian Fraboul, Abdellah Ait Ouahman. "Virtual Walled-Garden Model for IMS Services Provisioning". 3rd edition of the National Security Days - JNS3. Rabat - Maroc, April 26 & 27, 2013.
  - Mohamed Maachaoui, Anas Abou El Kalam, Christian Fraboul, Abdellah Ait Ouahman. "Multi-Level Authentication Based Single Sign-On for IMS Services". 13th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security - CMS 2012, Canterbury, UK, 03/09/2012-05/09/2012, Springer, p. 174-187, September 2012.
  - Mohamed Maachaoui, Anas Abou El Kalam, Christian Fraboul. "A secure One-way authentication protocol in IMS Context". 12th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security - CMS 2011, Gent, Belgium, 19/10/2011-21/10/2011, Springer, p. 222-231, October 2011.
  - Mohamed Maachaoui, Anas Abou El Kalam, Christian Fraboul, Abdellah Ait Ouahman. "Amélioration du protocole d'authentification en environnement IMS". 6th Conf. on Network Architecture and Information System Security – SAR-SSI 2011.
  - Mohamed Maachaoui, Anas Abou El Kalam, Abdellah Ait Ouahman. "Model-Based Security Analysis for IMS Network". The International Conference on Multimedia Computing and Systems (ICMCS'11).
  - Anas ABOU EL KALAM, Mohamed MAACHAOUI, Noureddine IDBOUFKER, Hassan AIT LAHCEN, Abdellah AIT OUAHMAN. A secure architecture for SIP in the IMS context. NGNS'10, International Conference on Next Generation Network & Services.
  - Mahmoud MOSTAFA, Anas ABOU EL KALAM, Mohamed MAACHAOUI, Christien FRABOUL. "Q-ESP: Un nouveau protocole pour la sécurité et la QoS dans les Réseaux". Conference on Security in Network Architectures and Security of Information Systems - SAR-SSI 2010.

- **Articles de revues internationales :**
  - A. ABOU EL KALAM, M. MAACHAOUI, N. IDBOUFKER, H. AIT LAHCEN, A. AIT OUAHMAN. "A secure architecture for nomadic user in IMS Network". International Journal of Mobile Computing and Multimedia Communications (IJMCMC).
  - Mahmoud Mostafa, Anas Abou El Kalam, Mohamed Maachaoui, Noureddine Idboufker. Specification, Implementation and Performance Evaluation of the QoS-friendly Encapsulating Security Payload (Q-ESP) Protocol. WILEY, Security and Communication Network.

- **Contributions à des ouvrages de synthèse**
  - Contribution dans la réalisation d'un livre de synthèse dans le cadre du projet European Feel@Home. Titre: "Digital Home Network".

- **Rapports de projet Européen :**
  - Anas Abou El Kalam, Marta Bel, Olivier Dugeon, Marc Lacoste, Mohamed Maachaoui, Gema Maestro, Francisco Moyano, and Rodrigo Roman. “Security Architecture”-- Projet European Feel@Home -- Deliverable D5.1, Janvier 2010.
  - Marta Bel, Francisco Moyano, Rodrigo Roman, Mohamed Maachaoui, Anas Abou El Kalam, Marc Lacoste. “Authentication, Definition of Identity Management”-- Projet European Feel@Home -- Deliverable D5.3, Fevrier 2010.
  - Daniel Migault, Rodrigo Roman, Gema Maestro, Javier Rodríguez, Mohamed Maachaoui, Anas Abou el Kalam, Marc Lacoste, Marta Bel. “Network Security Architecture and Impact on the Network” -- Projet European Feel@Home -- Deliverable D5.4, Decembre 2010.

Références

[Aarn00] AARNES, ANDRE: *Public Key Certificate Revocation Schemes*, 2000

[ABGM09] AL-BEGAIN, KHALID ; BALAKRISHNA, CHITRA ; GALINDO, LUIS ANGEL ; MORO, DAVID: *IMS: A Development and Deployment Perspective*, 2009 — ISBN 9780470740347

[AJKL00] AARNES, ANDRE ; JUST, MIKE ; KNAPSKOG, SVEIN J. ; LLOYD, STEVE ; MEIJER, HENK: Selecting Revocation Solutions for PKI. In: *Fifth Nordic Workshop on Secure IT Systems (NORDSEC 2000)*, 2000

[Avis00] AVISPA PROJECT: *Automated Validation of Internet Security Protocols and Applications*. URL http://www.avispa-project.org/

[AZFL12] ARKKO, JARI ; ZORN, GLEN ; FAJARDO, VICTOR ; LOUGHNEY, JOHN: Diameter Base Protocol, RFC 6733 (2012)

[BEKL03] BOX, D ; EHNEBUSKE, D ; KAKIVAYA, G ; LAYMAN, A ; MENDELSOHN, N ; NIELSEN, H F ; THATTE, S ; WINER, D: Simple object access protocol (soap) 1.1, May 2000. In: *W3C Note* (2003) — ISBN 8

[Bert07] BERTRAND, GILLES: The IP Multimedia Subsystem in Next Generation Networks. In: *Architecture* Bd. 7 (2007)

[Boun00] BOUNCYCASTLE: *BouncyCastle*. URL http://www.bouncycastle.org

[BuDP06] BURR, WILLIAM E ; DODSON, DONNA F ; POLK, W TIMOTHY: *NIST Electronic Authentication Guideline*. Bd. 43, 2006 — ISBN 1470110725, 9781470110727

[Cant05] CANTOR, S., HIRSCH, F., KEMP, J., PHILPOTT, R., AND E. MALER: *„Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0“, OASIS Standard saml-bindings-2.0-os*, 2005

[Cbek04] C. BEKARA, M. LAURENT-MAKNAVICIUS: *„Méthodes de révocation des certificats numériques“, Rapport de recherche du GET*, 2004

[Chen06] CHEN, E.Y.: Detecting DoS attacks on SIP systems. In: *1st IEEE Workshop on VoIP Management and Security, 2006.* (2006) — ISBN 1-4244-0144-5

[Cipa00] CIPANGO: *cipango | SIP/HTTP Servlets Application Server*. URL http://www.cipango.org/. - abgerufen am 2014-12-04

[CLGZ03] CALHOUN, P ; LOUGHNEY, J. ; GUTTMAN, E. ; ZORN, G. ; ARKKO, J.: Diameter Base Protocol. In: *IETF RFC 3588* (2003), S. 1–148

[DoYa81] DOLEV, D. ; YAO, A. C.: On the security of public key protocols. In: *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)* (1981) — ISBN 0018-9448

[East06] EASTLAKE. D, HANSE. T: *RFC 4634: US Secure Hash Algorithms (SHA and HMAC-SHA)*. URL http://www.hjp.at/doc/rfc/rfc4634.html. - abgerufen am 2014-11-15

[Etsi00] ETSI/TISPAN: *EuropeanTelecommunications Standards Institute / Telecoms & Internet converged Services & Protocols for Advanced Networks*. URL http://www.etsi.org/tispan/

[Etsi11] ETSI: *European Telecommunications Standards Institute (ETSI) TS 187 003 - V3.4.1 - Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Security; Security Architecture.* URL https://archive.org/details/etsi_ts_187_003_v03.04.01. - abgerufen am 2014-11-17

[Fran03] FRANKEL. S, GLENN. R, KELLY. S: *RFC 3602: The AES-CBC Cipher Algorithm and Its Use with IPsec*. URL http://www.hjp.at/doc/rfc/rfc3602.html. - abgerufen am 2014-11-15

[Gonz00] GONZALO CAMARILLO, MIGUEL-ANGEL GARCÍA-MARTÍN: *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds, 3rd Edition*. URL http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470516623.html. - abgerufen am 2014-11-16

[Gpp00] 3GPP: *3GPP : 3rd Generation Partnership Project*. URL http://www.3gpp.org/

[Gpp05] 3GPP: *3GPP TS 33.222 - GAA: Access to NAFs using HTTPS*. URL http://www.in2eps.com/3g33/tk-3gpp-33-222.html. - abgerufen am 2014-11-17

[Gpp07] 3GPP: *3GPP TS 29.229. Technical Specification Core Network; Cx and Dx Interfaces Based on the Diameter Protocol; Protocol Details.*, 2007

[Gpp08] 3GPP: *3GPP TS 23.060: „General Packet Radio Service (GPRS)"; Service description; Stage 2.*, 2008

[Gpp09a] 3GPP: *3GPP TS 33.203. Technical Specification Group Services and Systems Aspects; 3G Security; Access security for IP based services (Release 9). v9.3.0.*, 2009

[Gpp09b] 3GPP: *3GPP TS 33.210: „3G security; Network Domain Security (NDS); IP network layer security". V8.3.0 s.l.: ETSI.*, 2009

[Gpp09c] 3GPP: *3GPP TS 33.102: „Security architecture". V8.4.0*, 2009

[Gpp09d] 3GPP: *3GPP TS 33.105: „Cryptographic algorithm requirements". s.l. : ETSI, Vol. 8.*, 2009

[Gpp10] 3GPP: *3GPP TS 29.228. Technical Specification Core Network; IP Multimedia Subsystem Cx and Dx Interfaces; Signaling Flows and Message Contents (Release 5).*, 2010

[Gpp12] 3GPP: *3GPP TS 33.220, Technical Specification Group Services and System Aspects, Generic Authentication Architecture (GAA), Generic Bootstrapping Architecture (GBA), (Release 11), V11.1.0*, 2012

[HaPJ06] HANDLEY, MARK ; PERKINS, COLIN ; JACOBSON, VAN: SDP: Session Description Protocol. RFC 4566 (2006)

[Hard12] HARDT, DICK: *The OAuth 2.0 Authorization Framework*, 2012

[Herz05] HERZOG, JONATHAN: A computational interpretation of Dolev-Yao adversaries. In: *Theoretical Computer Science* Bd. 340 (2005), Nr. 1, S. 57–81

[Hloc08] H LOCKHART, B CAMPBELL, N RAGOUZIS: *Security Assertion Markup Lnaguage (SAML) V2.0 Technical Overview*. URL https://www.oasis-open.org/committees/download.php/14360/sstc-saml-tech-overview-2.0-draft-08-diff.pdf. - abgerufen am 2014-11-15

[Htsc10] H. TSCHOFENIG, J. PETERSON, J. POLK, D. SICKER, AND J. HODGES: *"SIP SAML Profile and Binding", status: IETF Draft Standard*, 2010

[HuLi07] HUANG, CHUNG MING ; LI, JIAN WEI: One-pass authentication and key agreement procedure in IP multimedia subsystem for UMTS. In: *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2007 — ISBN 0769528465, S. 482–489

[Inte03] INTERNATIONAL TELECOMMUNICATION UNION, TELECOMMUNICATION SECTOR, STANDARDIZATION: *„Security architecture for systems providing end-to-end communications", ITU-T Rec.X.805*, 2003

[IsGr09] ISLAM, SALEKUL ; GRÉGOIRE, JEAN-CHARLES: User-centric service provisioning for IMS. In: *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems - Mobility '09*. New York, New York, USA : ACM Press, 2009 — ISBN 9781605585369, S. 1–8

[IsGr11] ISLAM, SALEKUL ; GRÉGOIRE, JEAN CHARLES: Multi-domain authentication for IMS services. In: *Computer Networks* Bd. 55 (2011), S. 2689–2704 — ISBN 1389-1286

[Jean09] JEAN-CHARLES GREGOIRE, SALEKUL ISLAM: *An SSO-Enabled Architecture for Beyond the IMS Domain Services*. URL http://libra.msra.cn/Publication/13584201/an-sso-enabled-architecture-for-beyond-the-ims-domain-services. - abgerufen am 2014-11-17

[JePW02] JENNINGS, CULLEN ; PETERSON, JON ; WATSON, MARK: Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, RFC 3325 (2002)

[Jonp04] JON PETERSON: S/MIME Advanced Encryption Standard (AES), RFC 3853 (2004)

[Kbek10] K. BEKARA, M. LAURENT: „ La vie privée dans les environnements fédérés “. In: *Atelier Protection de la vie privée, Annecy.*, 2010

[Kent05] KENT, S: IP Encaplsulating Security Payload (ESP_. In: *RFC 4303* (2005), S. 1–44

[KeSe05a] KENT, S ; SEO, K: RFC 4301 Security Architecture for IP. In: *IETF* (2005), S. 1–101

[KeSe05b] KENT, S. ; SEO, K.: RFC4301: Security Architecture for Internet Protocol (IPSec). In: *Request for Comments, IETF* (2005), S. 1–102

[KnMT05] KNIGHTSON, KEITH ; MORITA, NAOTAKA ; TOWLE, THOMAS: NGN architecture: Generic principles, functional architecture, and implementation. In: *IEEE Communications Magazine* Bd. 43 (2005), Nr. 10, S. 49–56 — ISBN 0163-6804 VO - 43

[LCHW05] LIN, YI BING ; CHANG, MING FENG ; HSU, MENG T. ; WU, LIN Y.: One-pass GPRS and IMS authentication procedure for UMTS. In: *IEEE Journal on Selected Areas in Communications* Bd. 23 (2005), S. 1233–1239

[Libe06] LIBERTY ALLIANCE: *“Liberty Alliance Project: Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification’’, Version: v2.0*, 2006

[Litt00] LITTLEIMS: *Little IMS*. URL https://cipango.atlassian.net/wiki/display/LITTLEIMS/Home

[MAFA11] MAACHAOUI, M. ; ABOU EL KALAM, A. ; FRABOUL, C. ; AIT OUAHMAN, A.: Enhanced authentication protocol in IMS environment. In: *2011 Conference on Network and Information Systems Security, SAR-SSI 2011, Proceedings*, 2011

[MaGl98] MADSON, C. ; GLENN, R.: The Use of HMAC-SHA-1-96 within ESP and AH (1998)

[MaKO11] MAACHAOUI, M. ; KALAM, A. A E ; OUAHMAN, A. AIT: Model-based security analysis for IMS network. In: *International Conference on Multimedia Computing and Systems -Proceedings*, 2011

[MeFa04] MEALLING, MICHAEL ; FALTSTROM, PATRIK: The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM). In: *RFC 3761* (2004)

[Miik09] MIIKKA POIKSELKÄ, GEORG MAYER: *The IMS: IP Multimedia Concepts and Services, 3rd Edition*. URL http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470721960.html. - abgerufen am 2014-11-17

[MKFO13] MAACHAOUI, MOHAMED ; EL KALAM, ANAS ABOU ; FRABOUL, CRISTIAN ; OUAHMAN, ABDELLAH AIT: Virtual walled-garden model for IMS services provisioning. In: *2013 National Security Days (JNS3)* : IEEE, 2013 — ISBN 978-1-4799-0324-5, S. 1–6

[Moha11] MOHAMED MAACHAOUI, ANAS ABOU EL KALAM, CHRISTIAN FRABOUL. ; DE DECKER, B. ; LAPON, J. ; NAESSENS, V. ; UHL, A. (Hrsg.): *A Secure One-Way Authentication Protocol in IMS Context*, *Lecture Notes in Computer Science*. Bd. 7025. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011 — ISBN 978-3-642-24711-8

[Ndea00] N. DEASON: "SIP and SOAP," status: IETF Draft Standard (2000)

[NTTH09] NIE, PIN ; TAPIO, JUHA MATTI ; TARKOMA, SASU ; HEIKKINEN, JANI: Flexible single sign-on for SIP: Bridging the identity chasm. In: *IEEE International Conference on Communications*, 2009 — ISBN 9781424434350

[NtXS10] NTANTOGIAN, CHRISTOFOROS ; XENAKIS, CHRISTOS ; STAVRAKAKIS, IOANNIS: A generic mechanism for efficient authentication in B3G networks. In: *Computers and Security* Bd. 29 (2010), S. 460–475

[Oasi00] OASIS: *Advancing Open Standards for the Information Society*. URL https://www.oasis-open.org/org

[Oasi09] OASIS: SAML V2.0 Identity Assurance Profiles,Version 1.0 Committee Draft 01 22 (2009)

[PPSH10] PETERSON, JON ; POLK, JAMES ; SICKER, DOUGLAS ; HODGES, JEFF ; TSCHOFENIG, HANNES: SIP SAML Profile and Binding (2010)

[QKAW09] QADEER, MOHAMMED A. ; KHAN, AFAQ H. ; ANSARI, JUNED A. ; WAHEED, SARIYA: IMS network architecture. In: *Proceedings - 2009 International Conference on Future Computer and Communication, ICFCC 2009*, 2009 — ISBN 9780769535913, S. 329–333

[Rdro03] R. DROMS, J. BOUND, B. VOLZ, T. LEMON, C. PERKINS, AND M. CARNEY: Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, Internet Engineering Task Force (2003)

[Rdro97] R. DROMS.: DHCP: Dynamic Host Configuration Protocol. RFC 2131, Internet Engineering Task Force (1997)

[ReRe06a] RECORDON, DAVID ; REED, DRUMMOND: OpenID 2.0: A Platform for User-Centric Identity Management. In: *Discovery*, 2006 — ISBN 1595935479, S. 11–16

[ReRe06b] RECORDON, DAVID ; REED, DRUMMOND: OpenID 2.0. In: *Proceedings of the second ACM workshop on Digital identity management - DIM '06*, 2006 — ISBN 1595935479, S. 11

[Resc99] RESCORLA, E.: Diffie-Hellman Key Agreement Method (RFC 2631). In: *The Internet Society*, 1999

[ReSM08] REBAHI, YACINE ; SHER, MUHAMMAD ; MAGEDANZ, THOMAS: Detecting flooding attacks against IP multimedia subsystem (IMS) networks. In: *AICCSA 08 - 6th IEEE/ACS International Conference on Computer Systems and Applications*, 2008 — ISBN 9781424419685, S. 848–851

[RSCJ02] ROSENBERG, J ; SCHULZRINNE, H ; CAMARILLO, G ; JOHNSTON, A ; PETERSON, J ; SPARKS, R ; HANDLEY, M ; SCHOOLER, E: SIP: Session Initiation Protocol. RFC 3261. In: *Internet RFCs* (2002)

[SCFJ03] SCHULZRINNE, H ; CASNER, S ; FREDERICK, R ; JACOBSON, V: *RFC 3550: RTP: A Transport Protocol for Real-Time Applications*. . — IETF

[ShCM08] SHER, MUHAMMAD ; CARVALHO DE GOUVEIA, FABRICIO ; MAGEDANZ, THOMAS: IP Multimedia Subsystem (IMS) for Emerging All-IP Networks. In: FREIRE, M. ; PEREIRA, M. (Hrsg.) *IEncyclopedia of Internet Technologies and Applications. GI Global*, IGI Global (2008) — ISBN 9781591409939

[Shib00] SHIBBOLETH: *Shibboleth (Internet2)*. URL http://shibboleth.internet2.edu/

[ShMa06] SHER, MUHAMMAD ; MAGEDANZ, THOMAS: Secure access to IP multimedia services using generic bootstrapping architecture (GBA) for 3G & beyond mobile networks. In: *Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks - Q2SWinet '06* (2006), S. 17 — ISBN 1595934863

[ShMa07] SHER, MUHAMMAD ; MAGEDANZ, THOMAS: Protecting IP Multimedia Subsystem (IMS) service delivery platform from time independent attacks. In: *Proceedings - IAS 2007 3rd Internationl Symposium on Information Assurance and Security*, 2007 — ISBN 0769528767, S. 171–176

[Stur11] S. TURNER, T. POLK: RFC 6176 - Prohibiting Secure Sockets Layer (SSL) Version 2.0 (2011)

[Suda03] SUDA, BRIAN: SOAP Web Services. In: *Retrieved June* (2003)

[WaLi09] WANG, DONG ; LIU, CHEN: Model-based vulnerability analysis of IMS network. In: *Journal of Networks* Bd. 4 (2009), S. 254–262

[Wire00] WIRESHARK: *Wireshark*. URL http://www.wireshark.org

[YBDC06] YAHIA, IMEN GRIDA BEN ; BERTIN, EMMANUEL ; DESCHREVEL, JEAN PIERRE ; CRESPI, NOEL: Service definition for next generation networks. In: *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies,ICN/ICONS/MCL'06*. Bd. 2006, 2006 — ISBN 0769525520

[YiYH09] YING, NIU ; YAO, ZHAO ; HUA, ZOU: The study of multi-level authentication-based single sign-on system. In: *Proceedings of 2009 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IEEE IC-BNMT2009*, 2009 — ISBN 9781424450053, S. 448–452

[Zhen03] ZHENG, PEIFANG: Tradeoffs in certificate revocation schemes. In: *ACM SIGCOMM Computer Communication Review* Bd. 33, ACM (2003), Nr. 2, S. 103